International Journal of Innovative Research in Computer Science & Technology (IJIRCST)
ISSN: 2347-5552, Volume-2, Issue-3, May 2014

Minimizing Keystrokes/Clicks In Traditional Desktop Interfaces: Introducing A Modality-Diversification Layer

Anshuumaan Dwivedi, Ankit Goyal, Kartik Sharma and Kakoli Banerjee

***Abstract*—. Desktop software applications are built with traditional user interfaces in mind. Unlike modern Smartphones and tablets where Tactile and other haptic user-interface modalities are primary, desktop user-interfaces put excessive focus on keystrokes/mouse clicks. It can be established from data that the leading cause of repetitive strain injuries and ensuing lost workdays at workplaces is carpal tunnel syndrome arising from keyboard/mouse usage. The leading focus of this study is to demonstrate a high level user interface layer that can coordinate between a multiplicity of modalities and can be used to drastically reduce the number of mouse-clicks/keystrokes. The modalities will be implemented bottom up and the algorithms involved will be detailed. The user interface layer can be connected to any application through a simple API , thus it can be used to extend the accessibility of any existing software.**

***Index Terms*— Voice-actuated-events, Phonemes, Speech-synthesis, Classifier, Bright-spot Detection, Nose tip tracking, Eye-Blink-detection.**

INTRODUCTION

The input modalities in Computing Environments usually consist of Mouse, Keyboard, Webcam, Microphone.

In the presence of multiple input modalities there is no point in restricting ourselves to Mouse/Keyboard and use them disproportionately. This disproportionate propensity to use the two at the cost of other input devices like the camera and the microphone comes at a certain ergonomic expense to the user. In addition to the physical overhead of excess keystrokes and mouse movements for simple operations , it also reduces accessibility for differently-abled users. There is a need for an integrated solution that employs a synergy of non-keyboard/mouse input devices to smoothly operate traditional user interfaces.

Manuscript received May 22, 2014

Anshuumaan Dwivedi, JSS Academy of Technical Education,Noida,India(anshuumaan@hotmail.com,)

Ankit Goyal, JSS Academy of Technical Education,Noida,India(iamankitgoyal@hotmail.com)

Kartik Sharma, JSS Academy of Technical Education,Noida,India(kartiksharma1891@yahoo.in

Kakoli Banerjee, JSS Academy of Technical Education, Noida, India

With the availability of high speed processors and inexpensive webcams/microphones, more and more people have become interested in real-time applications that involve image processing/speech recognition. One of the promising fields in artificial intelligence is HCI which aims to use human features (e.g. face, hands, voice) to interact with the computer. One way to achieve that is to capture the desired feature with a webcam and monitor its action in order to translate it to some events that communicate with the computer.

In our work we were trying to compensate hand movements with an application that uses facial features (nose tip and eyes) and human voice to interact with the computer. The nose tip was selected as the pointing device; the reason behind that is the location and shape of the nose; as it is located in the middle of the face it is more comfortable to use it as the feature that moves the mouse pointer and defines its coordinates, not to mention that it is located on the axis that the face rotates about, so it basically does not change its distinctive convex shape which makes it easier to track as the face moves. Eyes were used to simulate mouse clicks, so the user can fire their events as he blinks. While different devices were used in HCI (e.g. infrared cameras, sensors, microphones) we used an off-the-shelf webcam that affords a moderate resolution and frame rate as the capturing device in order to make the ability of using the program affordable for all individuals. We will try to present an algorithm that distinguishes true eye blinks from involuntary ones, detects and tracks the desired facial features precisely, and fast enough to be applied in real-time.

MOTIVATION

The primary motivation behind studying alternative routes of human computer interaction is the prevalence of RSI(repetitive strain injury) among regular users of keyboards and mice. Figure 1 illustrates the prevalence of a typical RSI, i.e. the Carpal Tunnel Syndrome, the etiology of which is sometimes traced to repetitive keyboard/mouse usage.[1]

FIGURE 1. Analysis on causes of lost workdays

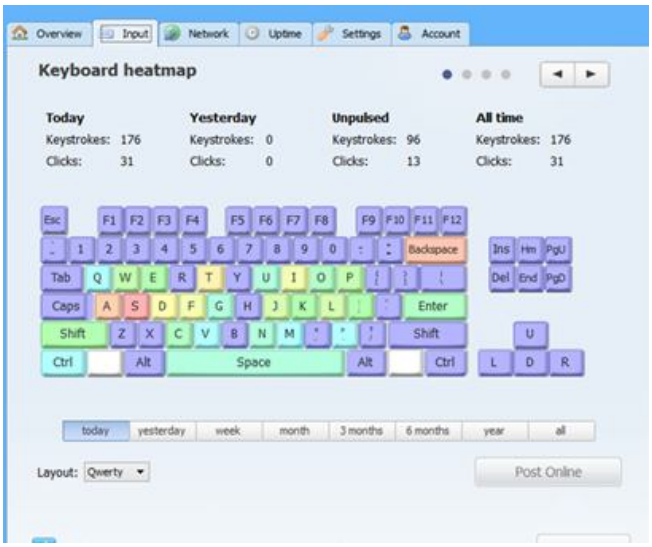

FIGURE 2. In the above figure a casual analysis of number of keystrokes for an average user is presented as a justification for this project.[9]

ARCHITECTURE OF THE SOLUTION

The system is a loosely coupled array of modules which have no lateral interaction. The common UI layer can be used by the user to shut down or start a particular module.

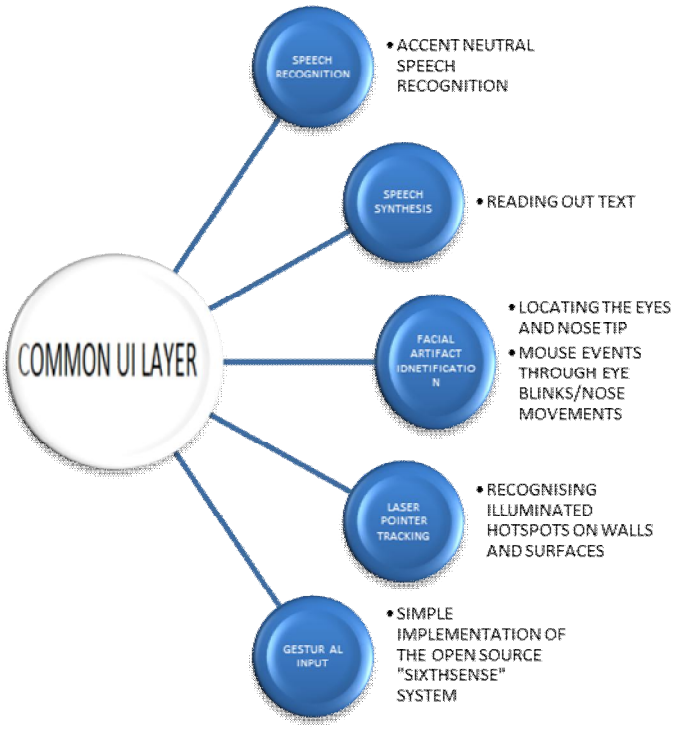

FIGURE 3. Modules of the system

I.

II. UNDERLYING COMPONENTS

FACE DETECTION

Face detection has always been a vast research field in the computer vision world, considering that it is the backbone of any application that deals with the human face (e.g. surveillance systems, access control).Researchers did not spare any effort or imagination in inventing and evolving methods to localize, extract, and verify faces in images. Early methods are dated back to 1970s [2], where simple heuristics were applied to images taken with certain restrictions (e.g. plain background, frontal view). These methods however have improved over time and become more robust to lighting conditions, face orientation, and scale.

Despite the large number of face detection methods, they can be organized in two main categories: Feature-based methods, and image-based methods[2].

1. The first involves finding facial features (e.g. nose trills, eye brows, lips, eye pupils….) and in order to verify their authenticity performs geometrical analysis of their locations, areas, and distances from each other. This feature-based analysis will eventually lead to the localization of the face and the features that it contains. Some of the most famous methods that are applied in this category are skin models, and motion cues which are effective in image segmentation and face extraction. On one hand feature-based analysis is known for its

pixel-accuracy features localization, and speed, on the other hand its lack of robustness against head rotation and scale has been a drawback of its application in computer vision.2. The second is based on scanning the image of interest with a window[2] that looks for faces at all scales and locations. This category of face detection implies pattern recognition, and achieves it with simple methods such as template matching or with more advanced techniques such as neural networks and support vector machines. Image-based detection methods are popular because of their robustness against head rotation and scale, despite the fact that the exhaustive window scanning requires heavy computations. More and more new detection methods are added to the arsenal of computer vision researchers, which proves once again the importance of this field and its ability of acquiring new ideas. Before over viewing the face detection algorithm that was applied in this work, here is an explanation of some of the idioms that are related to it.[2]

SSR FILTER

SSR Filter stands for: Six Segmented Rectangular filters. At the beginning, a rectangle is scanned throughout the input image. This rectangle is segmented into six Segments as shown in the figure below. In the next figure the filter is applied on a human face.

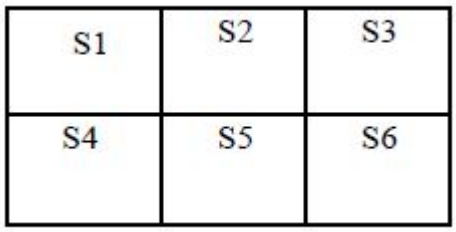

FIGURE 4: SSR Filter.

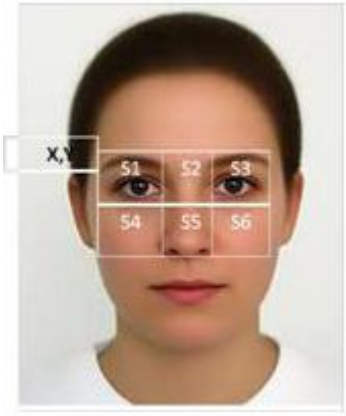

FIGURE 5: SSR on a face.

We denote the total sum of pixel value of each segment (S1-S6). The proposed SSR filter is used to detect the Between-the-Eyes[BTE] based on two characteristics of face geometry. (1) The nose area (S_n) is brighter than the right and left eye area (eye right (S_{er}) and eye left (S_{el}), respectively).

The following equations describe the relationships.

$S_n = S2 + S5$

$S_{er} = S1 + S4$

$Sel = S3 + S6$

Then,

$S_n > S_{er}$ (1)

$S_n > S_{el}$ (2)

$S_e < S_c$ (3)

When expression (1), (2), and (3) are all satisfied, the center of the rectangle can be a candidate for Between-the-Eyes The sum of pixels2 in each sector is denoted as S along with the sector number [2] even these filter sizes do not completely contain both eyes area, because only some parts of eyes are still darker than nose area.[3]

INTEGRAL IMAGE

In order to facilitate the use of SSR filters an intermediate image representation called integral image will be used. The concept has been borrowed from Viola and Jones at Mitsubishi. (Please refer to [7]) In this representation the integral image at location x, y contains the sum of pixels which are above and to the left of the pixel x, y [7] Throughout this work, pixels' values are the grayscale values. With this representation calculating the sectors of the SSR filter becomes fast and easy. No matter how big the sector is, we will need only 3 arithmetic operations to calculate the sum of pixels which belong to it. Sector = D – B – C + A. So each SSR filter requires 6*3 operations to calculate it.

For the original image i(x, y), the integral image is defined as [3]

$$ii(x,y) = \sum_{x' \le x} \sum_{y' \le y} i(x', y') \qquad (1)$$

The integral image can be computed in one pass over the original image by the following pair of recurrences.[3]

$s(x, y) = s(x, y - 1) + i(x, y)$ (2)

$ii(x, y) = ii(x - 1, y) + s(x, y)$ (3)

Where $s(x, y)$ is the cumulative row sum, $s(x, -1) = 0$, and $ii(-1, y) = 0$.

Using the integral image, the sum of pixels within rectangle D (rs) can be computed at high speed with four array references as shown in Fig.1.

$$sr = (ii(x, y) + ii(x - W, y - L)) - (ii(x - W, y) + ii(x, y - L)) \quad (4)$$ [3]

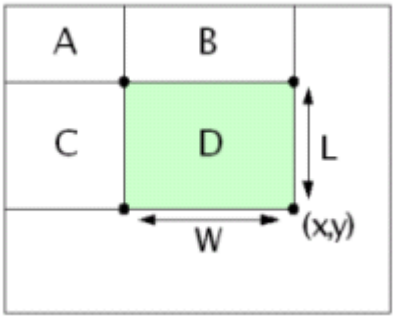

FIGURE 1. Integral Image

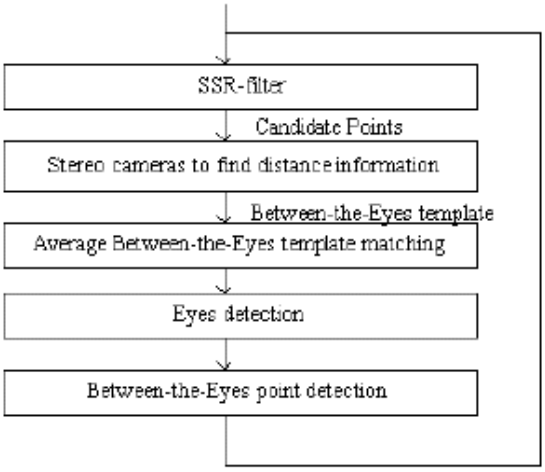

FIGURE 6. Processing Flow of Real-Time Face Detection

SVM THEORY

SVM stands for: Support Vector Machines, which are a new type of maximum margin classifiers. In 'learning theory' there is a theorem stating that in order to achieve minimal classification error the hyperplane which separates positive samples from negative ones should be with the maximal margin of the training samples [4], and this is what the SVM is all about. SVM takes as an input training data samples, where each sample consists of attributes and a class label (positive or negative). The data samples that are closest to the hyperplane are called support vectors. The hyperplane is defined by balancing its distance between positive and negative support vectors in order to get the maximal margin of the training data set. [4]

SPHINX4 SPEECH RECOGNITION FRAMEWORK

Sphinx-4 is a flexible, modular and pluggable framework to help foster new innovations in the core research of hidden Markov model (HMM) recognition systems. The design of Sphinx-4 is based on patterns that have emerged from the design of past systems as well as new requirements based on areas that researchers currently want to explore. To exercise this framework, and to provide researchers with a "research-ready" system, Sphinx-4 also includes several implementations of both simple and state-of-the-art techniques. The framework and the implementations are all freely available via open source.[8]

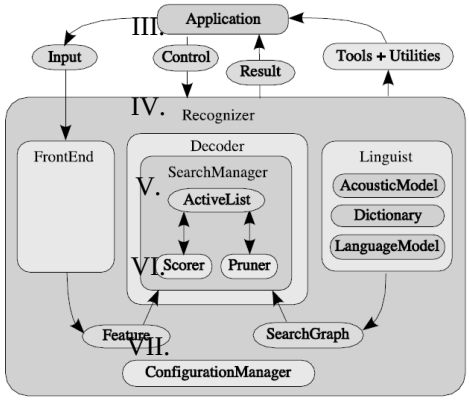

FIGURE 6: Sphinx 4 system Architecture.

ARCHITECTURE OF THE COMPONENTS

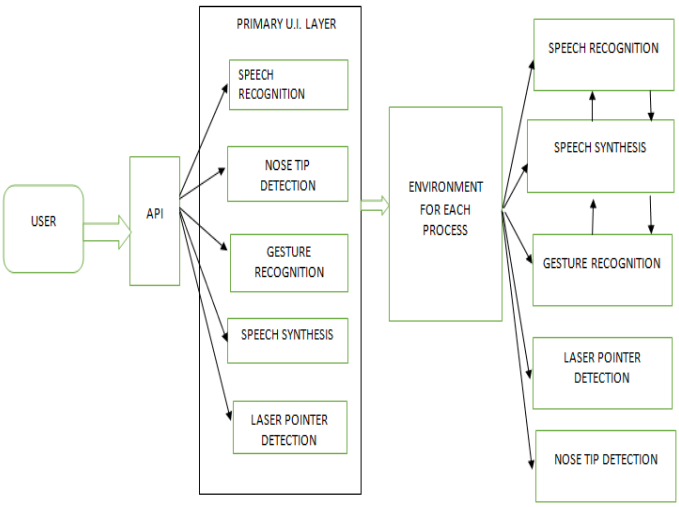

FIGURE 7: Interaction between components

IX. THE USER CAN USE A SIMPLE API TO PLUG ANY SOFTWARE INTO THE SYSTEM. THE API PROVIDES ACCESS TO THE PRIMARY U.I. LAYER WHICH USES BATCH FILES AND SIMPLE SYSTEM COMMANDS TO SWITCH BETWEEN DIFFERENT MODALITIES. SALIENT FEATURES OF THE SYSTEM ARE LISTED HEREUNDER:

1. The user can switch between the multiple modalities on his own or allow the system to "autopilot" the different modalities on the basis of the total keystrokes/clicks throughput.
2. The common environment is used to run all the modalities independently, there is minimal communication between the modalities.
3. Batch files are used to start and stop the invocation of modalities.

X. CONCLUSION

The new approach to improve accessibility developed in this effort is going to provide together with an overall reduction in the number of clicks/keystrokes, the ability to use multiple modalities simultaneously. In our work we are trying to compensate hand movements with an application that uses facial features (nose tip and eyes) and human voice to interact with the computer. The nose tip is selected as the pointing device. Eyes are used to simulate mouse clicks, so the user can fire their events as he blinks. While different devices are used in HCI (e.g. infrared cameras, sensors, microphones) we will use an off-the-shelf webcam that affords a moderate resolution and frame rate as the capturing device in order to make the ability of using the program affordable for all individuals. We will try to present an algorithm that distinguishes true eye blinks from involuntary ones, detects and tracks the desired facial features precisely, and fast enough to be applied in real-time.

Accent neutral Speech recognition, speech synthesis engine to read out screen text, facial artifacts identification from live webcam feed , locating the eyes and the nose tip ,nose tip movement for mouse pointer control, blink-actuated mouse events ,gestural input , laser pointer tracking: Recognizing illuminated hotspots on walls and surfaces are also implemented as the primary features of the system.

ACKNOWLEDGMENTS

This research paper has been made possible with the help and support of Mrs Kakoli Banerjee who was kind enough to read and finalize the paper and provided valuable suggestions. We owe her a special acknowledgment of gratitude for her support and encouragement.

REFERENCES

1."Image based Face Detection and Recognition: based Face Detection and Recognition: based Face Detection and Recognition: State of the Art" Faizan Ahmad, Aaima Najam and Zeeshan Ahmed ,department of Computer Science & Engineering, Beijing University of Aeronautics & Astronautics

2. Malachy J. Foley, University of North Carolina at Chapel Hill, NC " Avoiding Mouse Elbow"

3. Oraya Sawettanusorn, Yasutaka Senda, Shinjiro Kawato, Nobuji Tetsutani, and Hironori Yamauchi "REAL-TIME FACE DETECTION USING SIX-SEGMENTED RECTANGULAR FILTER (SSR FILTER)REAL-TIME FACE DETECTION USING SIX-SEGMENTED RECTANGULAR FILTER (SSR FILTER")

4. Chiang, C. C., Tai, W. K., Yang, M. T., Huang, Y. T. & Huang, C. J., (2003). A novel method for detecting lips, eyes and faces in real-time. Real-Time Imaging 9, 277-287.

5. Gurbuz, S., Kinoshita, K., & Kawato, S., (2004a).Real-time human nose bridge tracking in presence of geometry and illumination changes. Second International Workshop on Man-Machine Symbiotic Systems, Kyoto, Japan.

6. S. Kawato and J. Ohya. Two-step approach for real-time eye tracking with a new filtering technique. Proc. Int. Conf. on System , Man & Cybernetics, pages 1366–1371, 2000.

7. P.Viola and M.Jones, "Rapid object Detection using a Boosted Cascade of Simple Features," Proc. Of IEEE Conf.CVRP,1 , pp.511-518,2001.

8. Lamere, P.; Kwok, P.; Walker, W.; Gouva, E.; Singh, R.; Raj, B.; Wolf, P , MITSUBISHI ELECTRIC RESEARCH LABORATORIES"Design of the CMU Sphinx-4 Decoder"

9.http://www.addictivetips.com/windows-tips/whatpulse-for-windows/