TOUCH-FREE USER INTERFACE FOR AUGMENTED REALITY SYSTEMS

ANKUSH RAI*, JAGADEESH KANNAN R
School of Computing Science and Engineering, VIT University, Chennai, Tamil Nadu, India. Email: ankushresssci@gmail.com

ABSTRACT

Augmented reality is the upcoming field of research and is often suffer from the current form of user interface. In this study, we present touch-free user interactive system for augmented reality applications to carry out multi-task operations. We validated the efficacy of the method based on the performance of several users while carrying out the complex task in our sample augmented reality game.

Keywords: Multimedia, Interactive graphical user interface, Automation.

INTRODUCTION

In human-to-computer interaction, interacting for information about tangible objects within the computing media is usually accomplished by making indirect visual references to it. On the other hand, the human-to-human interaction with information is accomplished by the visual references which are addressed as the perception. Therefore, in this study, we meant to adjoin the two phenomenon of interaction adding depth to the graphical user interface (GUI) elements. Giving visual commands, finding information or issuing commands involving tangible objects can also be naturally accomplished by making similar visual references as found appropriated by the user. Unlike in system dependent on command line interface (CLI), a GUI has several test operations to successful pass on [1]. Thus, the testing techniques mainly employed for testing CLI programs suffer from scaling problems such as finite state machine when applied in the world of GUI's [2,3]. Therefore, it is required that a different approach is to be used for testing GUI's from what it is employed for CLI technique [4], which in turn involves usage of a planning system [5,6]. Although employing the technique of capture-playback, functions well in CLI world but often are prone to problems which are quite significant when it is implemented for a GUI system [7]. Accordingly, to eliminate such problems, there were two ways mainly used by the testers; either the testers use data collected from GUI interaction through the underlying windowing system [8], or to build the driver into the GUI such that the commands or events can be dispatched from other programs [9].

However, in the past, there is a field namely gesture control attempts made to achieve this through various computer vision algorithm over various platforms but requires hefty computing even for a simple job of identifying the gesture from real world to the trained gesture recognition from artificial neural networks or various pattern recognition algorithms which limits its real-time application. In some cases, few device manufacturing companies have developed many-core mobile computing devices where such gesture control features are provided; even though, it is not a cost-effective method cost more battery usage with delayed interactive actions. Thus, there is a need to eliminate such limitations; therefore, we have used the attribute based level adaptive thresholding algorithm for object extraction (ABLATA) [10]; which is implemented in augmented form with the proposed evolutionary algorithm to match up with the user defined motion inputs and classify to learn and imitate the process simultaneously.

MOBILE EVOLUTIONARY ALGORITHM FOR EVASIVE INTERACTIVE GUI

Algorithm

Input: Two images (a) Instance of window’s workspace W (b) instance of user’s end U

Output: Action sets ASi and matrix model of tree of actions Mi

Step 1: Perform ABLATA over U and W; such that, we get L_u and L_w which are the set of levels from U and W, respectively.

Step 2: Compute the pointing correlation state P as

$$P = \alpha \left[ \sum_{i=1}^{N_u} \sum_{j=1}^{N_w} \sum_{p_1}^{L_u} \sum_{p_2}^{L_w} S_{p_1,p_2} \left(f_1, f_2\right) \right]$$

where, $L_u$ are the universal set of level for the images, $p_1$ and $p_2$ are the adjacent pixels intersection with the levels $L_u$ and $L_w$ respectively, $S_{p1,p2}$ are the sets of pixel density constraint layout for the pixel positioning with its patterning saved in levels and between its intersection of adjoint pixels and the super positioned pixel density layout of differing state at the users instance of the image U. Furthermore, $t_i$ is the collection of patterns for the weighted superposed state P. (Initially its value is set to 0), $f_1, f_2$ are the two delay frames with a minimal time delay $t_i$ [11-13].

Step 3: For $p_{m_i}$ evaluate

$$P \rightarrow P_i$$

Step 4: Calculate the tree of action based on continuous feedback loop

$$\begin{bmatrix} P_1 \\ P_2 \\ P_3 \\ P_4 \\ P_5 \\ P_6 \end{bmatrix} = M X \begin{bmatrix} t_1 \\ t_2 \\ t_3 \end{bmatrix}$$

$$M = \begin{bmatrix} AS_1 & | & | \\ P_1 & = & P_4 \\ P_2 & = & P_5 \end{bmatrix} \begin{bmatrix} AS_2 & | & | \\ P_3 & = & P_6 \end{bmatrix} \begin{bmatrix} AS_3 & | & | \\ P_7 & = & P_8 \end{bmatrix} \ldots$$

$$AS_1$$

$$AS_2$$

$$AS_3$$

$$AS_4$$
Step 5: Repeat step 1-4, update $P_c$.

CONCLUSION

Fig. 1a gives the summarized analysis of the performance result for the testable complex operations (Represented by triangle, and order of execution by numeric 1 and 2, respectively) performed over the proposed GUI algorithm and data flow process (Fig. 1b). Note: $F$ is not testable operations. This paper presented and revealed a dynamically adaptive approach for automotive interaction with GUIs and offering various benefits and capabilities. We conclude by mentioning future offering by the proposed work.
Many users prefer a personalized experience we have successfully offered it with satisfaction and speed of performing tasks on computing device. This is the robust method to the present data and provides a advantageous utility to the users employed in environments in highly colonized work environments such as flight control and management unit, scientific laboratories, nuclear Power Stations and in Conferences or Academic/Business presentations. This automation is less affected by variations when users are entitled to work in crowded areas (as shown in Fig. 2). The operational performance is shown in Fig. 3. The users can save the template of the trained GUI and can forward it to their other mobile devices, such that this will be uninfluenced by the conversion function to map patterns between themes or to require users to normalize the execution environment by switching to the default workspace.

REFERENCES