Empirical Study of a Vision-based Depth-Sensitive Human-Computer Interaction System

Farzin Farhadi-Niaki
School of Electrical Engineering and Computer Science
University of Ottawa, Ottawa, Canada
ffarh101@uottawa.ca

Reza GhasemAghaei
School of Computer Science
Carleton University, Ottawa, Canada
rgaghaee@connect.carleton.ca

Ali Arya
School of Information Technology
Carleton University, Ottawa, Canada
arya@carleton.ca

ABSTRACT

This paper proposes the results of a user study on vision-based depth-sensitive input system for performing typical desktop tasks through arm gestures. We have developed a vision-based HCI prototype to be used for our comprehensive usability study. Using the Kinect 3D camera and OpenNI software library we implemented our system with high stability and efficiency by decreasing the ambient disturbing factors such as noise or light condition dependency. In our prototype, we designed a capable algorithm using NITE toolkit to recognize arm gestures. Finally, through a comprehensive user experiment we compared our natural arm gestures to the conventional input devices (mouse/keyboard), for simple and complicated tasks, and in two different situations (small and big-screen displays) for precision, efficiency, ease-of-use, pleasantness, fatigue, naturalness, and overall satisfaction to verify the following hypothesis: on a WIMP user interface, the gesture-based input is superior to mouse/keyboard when using big-screen. Our empirical investigation also proves that gestures are more natural and pleasant to be used than mouse/keyboard. However, arm gestures can cause more fatigue than mouse.

Author Keywords

3D; gesture interaction; HCI; usability; vision.

ACM Classification Keywords

H.5.2 [User Interfaces]: Input Devices and Strategies; I.3.6 [Methodology and Techniques]: Interaction Techniques.

General Terms

Experimentation; Human Factors; Measurement.

INTRODUCTION

While the field of Human-Computer Interaction (HCI) have always aimed to improve the interaction by making computers more practical and responsive to the user's requests, and minimizing the incompatibility between the human's cognitive model and the computer's ability to understand and respond properly [1], lately the research in HCI is showing a significant focus on creating interfaces that are more user-friendly, by applying natural communication and human skills in the user interface design. The new wave of input systems in video game consoles (such as Nintendo Wii, Xbox Kinect, and PlayStation Move) are examples of the trend toward a more "natural" interfaces, where computers adapt to human behavior rather than the other way around. Input/output techniques, interaction styles, and evaluation methods are the challenging fields of research in such gesture-based improvement [2].

With availability of in-expensive 3D cameras, many researchers have improved the quality of gesture-based systems by incorporating depth information as well as employing robust computer vision methods such as those provided by toolkits like OpenCV. On the other hand, a consolidated and reliable usability analysis has not been fulfilled for gesture-based input systems to see how and where they can be used. This paper is based on a prototype that combines a 3D camera with advanced vision software, and offers a novel study of usability of such system in performing common desktop tasks like accessing files, opening and resizing windows, etc. The study has considered a variety of factors such as complexity of tasks, screen size, and human factors like pleasantness, fatigue, and naturalness.

RELATED WORK

Technical

Recent studies have demonstrated that hand gesture systems are not only technical and theoretical in nature but are also very practical since they can be implemented into numerous types of application systems and environments. For example, Ahn et al. [3] developed a method for virtual environment slide show presentations.

Another example is the study by Jain [4], which describes a way to estimate hand poses for mobile phones that only have one pointing gesture based on a vision-based hand gesture approach. The sign language tutoring tool developed by Aran et al. [5] is also very practical because it

APCHI '12, August 28–31, 2012, Matsue-city, Shimane, Japan.

is designed to interact with users to teach them the fundamentals of sign language [6].

Several researchers have conducted similar studies in tracking, such as the Viola-Jones-based cascade classifier, which is typically used for face tracking in rapid image processing [7,8] and is regarded as more robust in pattern recognition against noise and lighting conditions [9]. Other researchers have shown that cascade classifiers can also be utilized to recognize hands and various parts of the human body [9,10,11,12,13].

In order to detect gestures, Marcel et al. [14] proposed a method of hand gesture recognition based on Input-Output Hidden Markov Models that track variations in the skin color of the human body. Similarly, Chen et al. [15] applied the hidden Markov model in training method to enable systems to detect hand postures, even though it is more complex than Cascade classifiers in training hand gestures.

A simple Human-Computer Interactive system that could detect predefined hand gestures for the numbers 0 to 6 was proposed by Liu et al. [6]. This system could better implement the Number Input Management in Word documents. The AdaBoost algorithm was revised and used to automatically recognize a user's hand from the video stream, which is based on Haar-like features as a representation of hand gestures. A Multi-class Support Vector Machine was employed to train and detect the hand gesture based on Hu invariant moments features and the Human Computer Conversation was then implemented for hand gesture interaction instead of a traditional mouse and keyboard.

The other research, by Yu et al. [17], proposes a hand gesture feature extraction method (with a dataset of 3500 images) that employs multi-layer perception. By binarizing the image and enhancing the contrast, the silhouette and distinct features of the hand are accurately and efficiently extracted from the image. The Gauss-Laplace edge detection approach has been utilized to get the hand edge. A feature vector that can recognize hand gestures is developed from combinational parameters of Hu invariant moment, hand gesture region and Fourier descriptor.

In above mentioned related works, accuracy and usefulness of gesture recognition software have remained a challenging issue. Noise, inconsistent lighting, items in the background, distinct features, and equipment limitations can also be named as the constraints associated with some of those image-based gesture recognition systems. Technological incompatibility may also cause difficulties in the general usage to match various image-based gesture recognition systems. For instance, a calibrating algorithm for one camera might not work properly for another different camera. Kinect camera uses some more stable methods and very useful techniques such as: background removal, image segmentation, depth and connectivity detection, and hand gesture recognition. Last but not least, Kinect also works well in an extensive variety of lighting conditions which itself helps in reducing the need for a high power of CPU. Having all these features enables Kinect to simulate a number of controllers properly. Using Kinect unit enables us to identify the depth of every single pixel in the frame and ultimately conserve the developing (no need for making samples and efforts in training, and testing sessions) and running time comparing to the learning-based traditional methods that have been used in the above mentioned related works. Moreover, applying a depth thresholding removes the wrist and its unwanted defects from the depth map, based on Z (creates a binary image). Cropping the wrist out of the frame can also help in improving accuracy. On the other hand, OpenNI and NITE secure the system with a high stability and efficiency by decreasing the effect of ambient disturbing factors such as noise and improper light conditions. In addition, programming with NITE provides some gesture detector options, e.g. Velocity or Angle features in a push detector in order to make a desirable setting for the push gesture recognition.

Usability

As for the multimodal interfaces, Cabral et al. [18] discuss numerous usability issues associated to the use of gestures as an input mode. A simplistically strong 2D computer vision based gesture recognition system was introduced by the authors and was successfully used for interaction in VR environments. Three different scenarios were employed to test the interface: as a regular pointing device in a GUI interface, as a navigation tool, and as a visualization tool. Their results illustrated that it is more time consuming, as well as more fatiguing to complete simple pointing tasks than using a mouse. However, several advantages are revealed by the use of gestures as a substitute in multimodal interfaces. These include immediate access to computing resources using a natural and intuitive way, and that balances properly to joint applications, where gestures can be used infrequently.

A proposition by Villaroman et al. [19] suggests that using Kinect to classroom training on natural user interaction creates a prospect and innovative method. Examples are presented to demonstrate how Kinect-assisted instruction can be utilized to accomplish certain learning results in Human Computer Interaction (HCI) courses. Moreover, the authors have confirmed that OpenNI, in addition to its accompanying libraries, are adequate and beneficial in enabling Kinect-assisted learning activities. For students, Kinect and OpenNI offer a hands-on experience with its gesture-based, natural user interaction technology.

In a study on 3D applications using Kinect, Kang et al. [20] introduced a control method that naturally regulates the application with the use of distance information and joints' location information. Furthermore, the recognition rate was more successful, as well as the use of the proposed gestures in the 3D application, which was 27% quicker than a mouse.

Code Space, introduced by Bragdon et al. [21], is a system that combines touch + air gesture hybrid interactions to jointly carry small developer group meetings. This method enables access, control and sharing of information through several different devices such as multi-touch screen, mobile touch devices, and Microsoft Kinect sensors. In a formative study, professional developers were positive about the interaction design, and most felt that pointing with hands or devices and forming hand postures are socially acceptable.

A gesture user interface application, Open Gesture, is available for standard tasks, for instance making telephone calls, operating the television, and executing mathematical calculations. This prototype uses a television interface to carry out various tasks by using simple hand gestures. Based on a usability evaluation, Bhuiyan and Picking [22], recommend that this technology can improve the lives of the elderly and the disabled users by creating more independence while some challenges still remain to be overcome.

During a study, on touch-free navigation through radiological images, analyzed by Ebert et al. [23], ten medical professionals tested the system by rebuilding a dozen images from a CT data. The experiment measured the response period and the practicality of the system compared to the mouse/keyboard control. An average of ten minutes was required for the participants to be at ease with the system. The response time was 120 ms, and the image recreation time using gestures was 1.4 time longer than using mouse/keyboard. However it does remove the potential for infection, for both patients and staff.

In a usability study, in order to have more accurate results, it is suggested to design a simple and minimalistic as possible simulated desktop interface with neutral colors to reduce user error or bias, while focusing on common desktop tasks to be relatively general. Moreover, we believe that studying more features in a usability study than those have been studied in above mentioned related works develops the models and theories of interaction.

METHODOLOGY

Our research aims as verifying the following hypothesis: On a WIMP (Windows, Icons, Menus, Pointers) user interface, the gesture-based input is superior to mouse/keyboard when using with a big-screen, but not on a small screen. In order to verify this hypothesis, we (1) designed a simple yet effective simulated WIMP interface, (2) defined a set of criteria for evaluation, (3) selected natural gestures, (4) implemented a gesture recognition engine, and (5) performed usability studies.

User Interface and Gesture Recognition Modules

This project uses a simulated WIMP interface. The design is kept as simple and minimalistic as possible, with neutral colors to reduce user error or bias (Figure 1). The simulated desktop includes icons with operations such as selecting, opening/closing, moving and resizing.

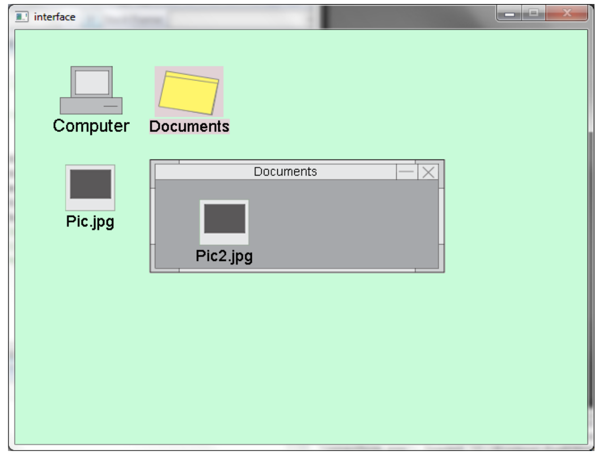


Figure 1. User interface.

Tables 1 and 2 show our chosen gestures and their corresponding mouse/keyboard events. We have used a combination of Kinect sensor, OpenCV, Allegro graphics library, OpenNI, and NITE to create the simulated desktop interface and interact with users.

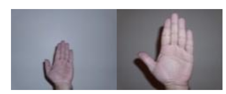
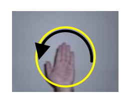
Processes	Arm
Selecting/Running/Closing	Hand pushing 
Moving curser	Hand moving 
Grabbing/Resizing	Hand circling 

Table 1. Final design for arm/hand set.

Arm Gestures	Mouse	Actions
Push	Dbl-click	Run/close objects
Circle + move + push	Drag & drop	Move/resize

Table 2. Arm gestures' definitions, mouse analogies, and actions.

The details of the gesture recognition engine are not within the scope of this paper that is focused on usability study. Figure 2 illustrates our gesture-based UI algorithm.

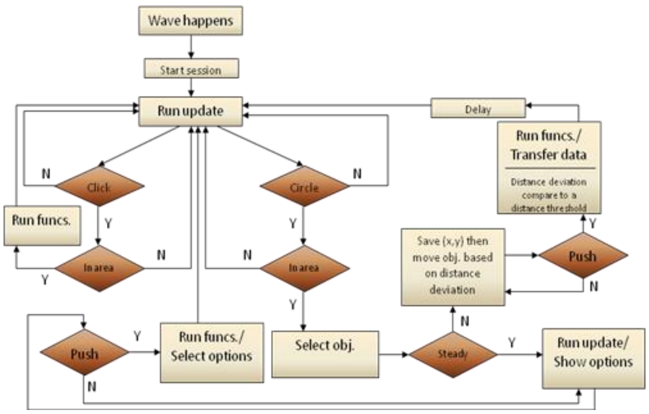


Figure 2. The algorithm controlling UI using arm gestures recognition.

User Experiments

In our usability experiments we have focused on common desktop tasks to be relatively general, and have included ratings by typical university users and also objective measures by observation, such as number of trials, errors, etc.

The experiment process has the following sessions:

Training session

Training consists of thirty minutes of practicing the "simple" tasks, including selecting desktop objects (icons and windows), opening and closing, moving and resizing. Complex tasks are combination of 5 simple ones through a script.

Test session

Test sessions include two tasks (simple and complex), two devices (mouse and gesture), and two types of screen (desktop and big-screen), i.e. eight units.

Questionnaire and observation

During the test sessions the users are requested to rate their satisfaction on a scale of 1 to 5 (1 for absolutely unsatisfied and 5 for extremely satisfied) on eight respective task tables, and to answer some extra questions on the questionnaire while the testing persons measure the observations.

RESULTS AND DISCUSSIONS

This study is conducted using 20 participants (10 males and 10 females) and in the age range of 11 to 40 (average of 29 years old). Nineteen participants were right-handed and one was left-handed.

Hypotheses and Analyses

For the different factors being studied, 3-way repeated analysis of variances (ANOVA) is carried out for three independent variables:

1- Difficulty (simple task vs. complex task)
2- Input device (mouse vs. arm gestures)
3- Output device (desktop vs. big-screen)

All analysis are concluded at $p < 0.05$ significance level and for 20 participants. Our ANOVA analysis is accompanied by an extra t-test analysis particularly for naturalness and fatigue. This redundancy is carried out in order to confirm our multi-factor analysis with a single-factor analysis. The results of the t-test support the ANOVA analysis.

Notation: In the following analyses, we show the mean and standard deviation for different variables in the forms of $M_{variable}$ (e.g. M_{simple} is the mean for simple task) and $SD_{variable}$ (e.g. $SD_{gesture}$ is the standard deviation for arm gestures). Moreover, F(df,MS) is the test statistic (F-ratio) in which df and MS are the degree of freedom and mean square respectively for the variables (within variables when more than one, and within subjects). The F-ratio is calculated using $MS_{variable(s)}/MS_{error(s)}$ and P is the probability value.

Time (duration of test session):

Hypothesis- using a mouse is faster than using arm gestures as inputs.

The analysis illustrates that for variable 1, $F(1,2504.306) = 66.994$, $P = 0.0000$ ($M_{simple} = 17.83$, $SD_{simple} = 7.67$ vs. $M_{complex} = 25.74$, $SD_{complex} = 9.80$). This illustrates that task complexity has significant effect on time. This effect is as expected since the two tasks were initially designed to illustrate different difficulty levels for using the system.

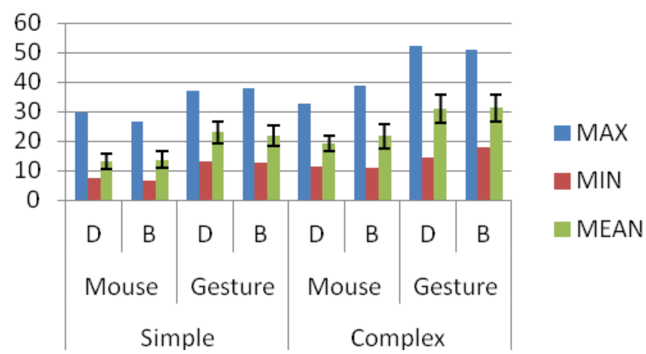


Figure 3. Temporal MAX/MIN/MEAN/ST DEV facts (D≡desktop, B≡big-screen).

For variable 2, $F(1,3820.070) = 41.163$, $P = 0.0000$ ($M_{mouse} = 16.90$, $SD_{mouse} = 7.0868$ vs. $M_{gesture} = 26.67$, $SD_{gesture} = 9.3749$), which implies that using gestures also has significant effect on time. For variable 3, $F(1,10.404) = 0.646$, $P = 0.4316$ which illustrates that the screen type does not have a significant effect on time. Moreover, the analysis shows no significant effect on time for variables 1 and 2 combined $F(1,29.929) = 1.371$, $P = 0.2562$, variables 1 and 3 combined $F(1,28.392) = 1.641$, $P = 0.2156$, and finally variables 2 and 3 combined, $F(1,37.056) = 1.131$, $P = 0.3008$. Combination of the three variables (1, 2, and 3) $F(1,0.121) = 0.006$, $P = 0.9370$ also do not show any significant effect on time. Based on the above, the initial hypothesis is confirmed meaning gesture inputs are significantly slower than using a mouse (as shown also in Figure 3).

Easiness (how easy to interact with the UI):

Hypothesis- Using arm gestures as inputs is easier than mouse.

Analyzing the feedback from participants regarding easiness of experiments given the 3 variables defined earlier shows that the only significant effect is caused by variable 2, F(1,19.600) = 23.059, P = 0.0001 (M_{mouse} = 4.3750, SD_{mouse} = 0.8325 vs. $M_{gesture}$ = 3.6750, $SD_{gesture}$ = 0.9517). This means that according to participants, the only variable with significant effect on easiness is the input device (mouse vs. gesture). For variable 1, F(1,0.100) = 0.134, P = 0.7181 and for variable 3, F(1,1.225) = 2.730, P = 0.1149. For combination of variables 1 and 2, F(1,0.100) = 0.409, P = 0.5303, variables 1 and 3, F(1,0.225) = 0.371, P = 0.5497, variables 2 and 3, F(1,4.225) = 4.219, P = 0.0540, and finally for variables 1, 2, and 3, F(1,0.225) = 0.609, P = 0.4449 which indicates that there is no significant effect. According to the provided statistics, the initial hypothesis is rejected which indicates that using a mouse is significantly easier than using arm gestures.

Fatigue (how fatiguing to interact with the UI):

Hypothesis- Using arm gestures produces more fatigue compared to mouse.

In this experiment the participants have been asked to rank higher if more fatigue is experienced. The feedback obtained from participants indicates that similar to easiness, variable 2 is the only one with significant effect F(1,45.156) = 31.813, P = 0.0000 (M_{mouse} = 1.4000, SD_{mouse} = 0.7730 vs. $M_{gesture}$ = 2.4625, $SD_{gesture}$ = 0.9929). This indicates that the input device is the only determining parameter in fatigue. For variable 1, F(1,1.406) = 3.065, P = 0.0961 and for variable 3, F(1,0.506) = 1.351, P = 0.2595 respectively. For combination of variables 1 and 2, F(1,0.006) = 0.015, P = 0.9050, variables 1 and 3, F(1,0.006) = 0.018, P = 0.8949, variables 2 and 3, F(1,0.756) = 0.657, P = 0.4276, and finally variables 1, 2, and 3, F(1,0.756) = 1.322, P = 0.2645. Based on the above mentioned figures, the initial hypothesis is approved, meaning arms gestures significantly causes more fatigue compared to using a mouse. Table 3 shows an extra t-test analysis for fatigue which supports the ANOVA analysis.

Phase	**Mean**		**p-value**
	Mouse	**Arm**	
Simple/Desktop	1.10	2.45	3.756e-06
Simple/Big-screen	1.5	2.3	0.002506
Complex/Desktop	1.45	2.50	0.002502
Complex/Big-screen	1.55	2.60	0.002173

Table 3. Fatigue and results of t-test.

Naturalness (how natural/intuitive to interact with the UI):

Hypothesis- Using arm gestures is more natural than using a mouse.

For this factor, none of the variables shows any significant effect.

The calculated statistical values for variable 1, F(1,0.000) = 0.000, P = 1.0000, for variable 2, F(1,10.000) = 4.153, P = 0.0557, and for variable 3, F(1,0.225) = 0.851, P = 0.3679.

These results indicate that variables 1, 2, and 3 do not have any significant impact on naturalness of tasks. However, combination of variables 2 and 3 show significant effect F(1,5.625) = 6.628, P = 0.0186 ($M_{mouse\text{-}desktop}$ = 3.4500, $SD_{mouse\text{-}desktop}$ = 1.1082, vs. $M_{mouse\text{-}bigscreen}$ = 3, $SD_{mouse\text{-}bigscreen}$ = 1.1983, vs. $M_{gesture\text{-}desktop}$ = 3.5750, $SD_{gesture\text{-}desktop}$ = 0.9306, vs. $M_{gesture\text{-}bigscreen}$ = 3.8750, $SD_{gesture\text{-}bigscreen}$ = 0.8530). This means that the input device when combined with a particular output device will show significant effect on naturalness. Multiple one-way ANOVAs further indicate that mouse when used on desktop is significantly more natural than mouse used on big-screen. Moreover, gestures used on big-screen are significantly more natural than mouse used on both desktop and big-screen. Combination of variables 1 and 2, F(1,0.400) = 0.910, P = 0.3520, variables 1 and 3, F(1,0.225) = 0.533, P = 0.4744, and finally variables 1, 2, and 3 , F(1,0.625) = 1.067, P = 0.3145, show no significant effect. According to the above mentioned figures, the hypothesis is rejected, meaning arm gestures as inputs do not feel significantly more natural compared to mouse. However, it is shown that using arm gestures on big-screen is significantly more natural than using a mouse on both the desktop and the big-screen. Table 4 shows an extra t-test analysis for naturalness which supports the ANOVA analysis.

Phase	**Mean**		**p-value**
	Mouse	**Arm**	
Simple/Desktop	3.30	3.65	0.2804
Simple/Big-screen	3.05	3.90	0.006697
Complex/Desktop	3.6	3.5	0.7647
Complex/Big-screen	2.95	3.85	0.01963

Table 4. Naturalness and results of t-test.

Pleasantness (how pleasant to interact with the UI):

Hypothesis- Using arm gestures as inputs is more pleasant than using mouse.

When analyzing the participant feedback for pleasantness, a similar trend to that of naturalness is observed. Variable 1, F(1,0.006) = 0.016, P = 0.9020, variable 2, F(1,6.806) = 3.824, P = 0.0654, and variable 3, F(1,0.506) = 1.351, P = 0.2595 show no significant effect. Combination of variables 1 and 2, F(1,1.056) = 3.055, P = 0.0966, variables 1 and 3, F(1,0.306) = 1.347, P = 0.2601, and variables 1, 2, and 3, F(1,0.506) = 1.572, P = 0.2251 show no significant effect as well. Similar to naturalness, the only set of variables which illustrate an effect are combination of factors 2 and 3, F(1,8.556) = 7.716, P = 0.0120 ($M_{mouse\text{-}desktop}$ = 3.7250, $SD_{mouse\text{-}desktop}$ = 0.9868 vs. $M_{mouse\text{-}bigscreen}$ = 3.1500, $SD_{mouse\text{-}}$

bigscreen = 1.0266, vs. $M_{gesture\text{-}desktop}$ = 3.6750, $SD_{gesture\text{-}desktop}$ = 0.8590, vs. $M_{gesture\text{-}bigscreen}$ = 4.0250, $SD_{gesture\text{-}bigscreen}$ = 0.8317). Therefore there is significant interaction between input and output device when pleasantness is being analyzed. Multiple one-way ANOVAs further indicate that mouse when used on desktop is significantly more pleasant than mouse used on big-screen. Furthermore, arm gestures used on big-screen is significantly more pleasant than mouse used on desktop, mouse used on big-screen, and arm gestures used on desktop.

Based on these results, similar to naturalness, the initial hypothesis is rejected. But again, it is revealed that the hypothesis does hold true on big-screens, meaning using arm gestures is significantly more pleasant than mouse when performed on big-screens. Also it is shown that arm gestures used on big-screen is significantly more pleasant compared to when it is used on desktop.

Overall Satisfaction (how overall satisfactory to interact with the UI):

Hypothesis- Overall, using arm gestures as inputs is a more popular experience compared to mouse.

In the overall ranking obtained from participants, no particular variable shows significant effect. This can be due to the fact that while some parameters such as naturalness are ranked higher for gesture on the big-screen, the fatigue level is increased at the same time. This experience, we believe leads to an overall insignificant ranking. The calculated values are as follows: For variable 1, $F(1,0.006) = 0.019$, $P = 0.8928$, for variable 2, $F(1,0.306) = 0.341$, $P = 0.5662$, and for variable 3, $F(1,0.306) = 0.721$, $P = 0.4063$. Similarly for combination of variables, no effect is observed since for variables 1 and 2, $F(1,0.156) = 0.704$, $P = 0.4120$, variables 1 and 3, $F(1,0.006) = 0.022$, $P = 0.8833$, variables 2 and 3, $F(1,3.906) = 4.249$, $P = 0.0532$, and finally for all three variables 1, 2, and 3, $F(1,0.006) = 0.035$, $P = 0.8531$. Based on this analysis, the hypothesis is rejected, meaning neither input hold a significant popularity over the other.

Hypotheses Verification

According to the provided statistical analyses, we summarize our hypotheses verification as follows:

The time and the fatigue factors analyses support our initial hypotheses, meaning gesture inputs are significantly slower and more fatiguing than using a mouse. The initial hypotheses for the easiness and overall satisfaction factors are rejected which indicate that using a mouse is significantly easier than using arm gestures while neither inputs hold a significant popularity over the other. For the naturalness and the pleasure factors, the hypotheses are rejected as well, meaning arm gestures as inputs do not feel significantly more natural or more fun to use compared to mouse. However, it is revealed that using arm gestures on big-screen is significantly more natural and more pleasant than using a mouse on both the desktop and the big-screen. Also it is shown that arm gestures used on big-screen is significantly more pleasant compared to when it is used on desktop.

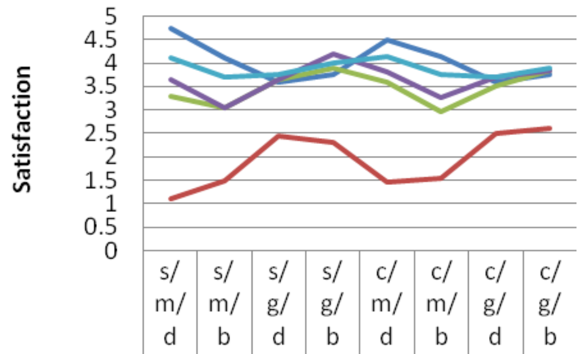
Extra Observations

Timing:

Using mouse on big-screen is slower than on desktop. As expected, due to not being familiar with controlling a UI using gestures, the result with mouse is faster than with gestures. However, we believe that having more practice and getting used to the gesture application, allows the users to perform the tasks almost as fast as using a mouse.

Satisfaction:

Most of the participants preferred "equally use of mouse and gesture" as a combination of gesture and mouse inputs.



	s/m/d	s/m/b	s/g/d	s/g/b	c/m/d	c/m/b	c/g/d	c/g/b
Easy	4.75	4.1	3.6	3.75	4.5	4.15	3.6	3.75
Fatigue	1.1	1.5	2.45	2.3	1.45	1.55	2.5	2.6
Natural	3.3	3.05	3.65	3.9	3.6	2.95	3.5	3.85
Pleasent	3.65	3.05	3.65	4.2	3.8	3.25	3.7	3.85
Overall	4.1	3.7	3.75	4	4.15	3.75	3.7	3.9

Figure 4. Satisfaction comparison (s≡simple, c≡complex, m≡mouse, g≡gesture, d≡desktop, b≡big-screen).

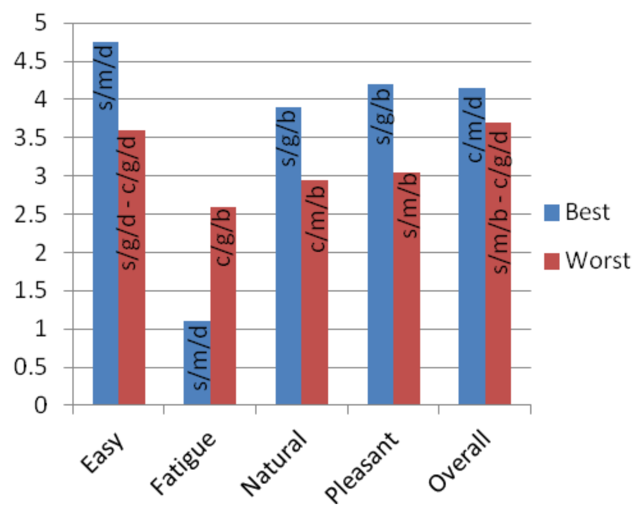


Figure 5. Best/Worst satisfactions (s≡simple, c≡complex, m≡mouse, g≡gesture, d≡desktop, b≡big-screen).

As shown in Figures 4 and 5, doing simple-task with gestures on desktop caused more fatigue than on big-screen, although it is reverse in doing complex-task. Performing simple-task, using mouse on desktop is the easiest and the lightest (least fatigue) and on big-screen is the least pleasant and the least overall satisfactory, while using gestures on big-screen is the most natural, and the most pleasant. In addition, the complex-task using gestures on desktop is the most difficult and the least overall satisfactory. In other words, a short time usage of mouse on big-screen, and a long term usage of gesture on desktop have the least popularity from users' feedback. Doing complex-task, using mouse on desktop is the most overall satisfactory and on big-screen is the least natural, while using gesture on big-screen is the heaviest (most fatigue).

Based on the results, opening a window (Running action) using gesture was the easiest task overall.

This study compared arm gestures with mouse/keyboard in two different settings (desktop and large-scale displays), and two different task difficulties (simple and complex). Based on the participants' feedback, multimodal UI makes more attentive and immersive than the conventional UI. There are still remaining issues to solve such that users feel fatigue while using arms in the air.

CONCLUSION

A new gesture-based interface has been presented and compared with traditional input systems for typical desktop tasks. Through an efficient implementation using Kinect 3D camera and computer vision software libraries, and with comprehensive user experiments, we compared our defined arm gestures to the conventional input devices (mouse/keyboard), in two different settings (desktop and big-screen displays), and during two sets of tasks (simple and complex) for precision, efficiency, easiness, pleasantness, fatigue, naturalness, and overall satisfaction to verify the following hypothesis: the gesture-based input is superior to mouse/keyboard when using big-screen. Our experiment has analytically showed that using gestures on a big-screen display is more natural and pleasant than using a mouse/keyboard in a HCI. On the other hand, arm gestures are more fatiguing than mouse.

There are a few efforts that can be undertaken to improve our prototype system. The current prototype only supports single hand gestures for interaction. Hence, multiple hands gesture interaction can be proposed in order to have more gestures available, reduce the error rate, and ultimately increase the accuracy, speed rate, and user satisfaction, while more hand postures will be selected to support the controlling activities. However, a robust approach in hand gesture recognition is necessary since the multiple hands increase the computational costs and complexity of the system. Using other types of body gestures and studying other types of tasks are among our objectives for further research.

ACKNOWLEDGMENTS

The authors wish to thank S. Ali Etemad for his great contribution in the usability part of this research, and also Colin Killby for his aid in designing our user interface.

REFERENCES

1. Harper, R., Rodden, T., Rogers, Y. and Sellen, A. Being human: Human-computer interaction in the year 2020. *Microsoft Corporation*, (2008).
2. Erol, A., Bebis, G., Nicolescu, M., Boyle, R.D. and Twombly, X. Vision-based hand pose estimation: A review. *Computer Vision and Image Understanding. Special Issue on Vision for Human-Computer Interaction.* vol. 108, (2007), 52-73.
3. Ahn, S.C., Lee, T.S., Kim, I.J., Kwon, Y.M. and Kim, H.G. Computer vision-based interactive presentation system. *Proceedings of Asian Conference for Computer Vision*, (2004).
4. Jain, G. Vision-based hand gesture pose estimation for mobile devices. *University of Toronto*, (2009).
5. Aran, O., Ari, I., Benoit, F., Campr, A., Carrillo, A.H., Fanard, P., Akarun, L., Caplier, A., Rombaut, M. and Sankur, B. Sign language tutoring tool. *eNTERFACE, The Summer Workshop on Multimodal Interfaces*, (2006).
6. Pang, Y.Y., Ismail, N.A. and Gilbert, P.L.S. A real time vision-based hand gesture interaction. *Fourth Asia International Conference on Mathematical/Analytical Modelling and Computer Simulation*, (2010), 237-242.
7. Viola, P. and Jones, M. Robust real-time object detection. *2nd International Workshop on Statistical and Computational Theories of Vision*, (2001).
8. Viola, P. and Jones, M. Rapid object detection using a boosted cascade of simple feature. *IEEE Computer Vision and Pattern Recognition*, vol. 1, (2001), 551-518.
9. Chen, Q., Cordea, M.D., Petriu, E.M., Varkonyi-Koczy, A.R. and Whalen, T.E. Human-computer interaction for smart environment applications using hand-gesture and facial-expressions. *International Journal of Advanced Media and Communication*, vol. 3 n.1/2, (2009), 95-109.
10. Kolsch, M. and Turk, M. Robust hand detection. *International Conference on Automatic Face and Gesture Recognition*, (2004).
11. Kolsch, M. and Turk, M. Analysis of rotational robustness of hand detection with a Viola-Jones detector. *In IAPR International Conference of Pattern Recognition*, (2004).
12. Zhang, Q., Chen, F. and Liu, X. Hand gesture detection and segmentation based on difference background image with complex background. *International Conference on Embedded Software and Systems*, (2008), 338- 343.
13. Anton-Canalis, L., Sanchez-Nielsen, E. and Castrillon-Santana, M. Hand pose detection for vision-based

gesture interfaces. *Conference on Machine Vision Applications*, (2005), 506-509.

14. Marcel, S., Bernier, O., Viallet, J.E. and Collobert D. Hand gesture recognition using input-output hidden Markov models. *Conference on Automatic Face and Gesture Recognition*, (2000).
15. Chen, F., Fu, C. and Huang, C. Hand gesture recognition using a real-time tracking method and hidden Markov models. *Image and Vision Computing*, (2003), 745-758.
16. Liu, Y. and Zhang, P. Vision-based human-computer system using hand gestures. *International Conference on Computational Intelligence and Security*, vol. 2, (2009), 529-532.
17. Yu, C., Wang, X., Huang, H., Shen, J. and Wu, K. Vision-based hand gesture recognition using combinational features. *Sixth International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, (2010), 543-546.
18. Cabral, M.C., Morimoto, C.H. and Zuffo, M.K. On the usability of gesture interfaces in virtual reality environments. *Proceedings of the 2005 Latin American conference on Human-computer interaction*, (2005), 100-108.
19. Villaroman, N., Rowe, D. and Swan, B. Teaching natural user interaction using OpenNI and the Microsoft Kinect sensor. *Proceedings of the 2011 conference on Information technology education*, (2011), 227-232.
20. Kang, J.W., Seo, D.J. and Jung, D.S. A study on the control method of 3-dimensional space application using Kinect system. *International Journal of Computer Science and Network Security*, vol.11, no.9, (2011), 55-59.
21. Bragdon, A., DeLine, R., Hinckley, K. and Morris M.R. Code space: Touch + air gesture hybrid interactions for supporting developer meetings. *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces*, (2011), 212-221.
22. Bhuiyan, M. and Picking, R. A gesture controlled user interface for inclusive design and evaluative study of its usability. *Journal of Software Engineering and Applications*, (2011), 513-521.
23. Ebert, L.C., Hatch, G., Ampanozi, G., Thali M.J. and Ross, S. You can't touch this: Touch-free navigation through radiological images. *Surg Innov Journal*, (2011).