Two-Handed Interactive Menu: An Application of Asymmetric Bimanual Gestures and Depth Based Selection Techniques

Hani Karam and Jiro Tanaka

Department of Computer Science, University of Tsukuba,
Tennodai, Tsukuba, 305-8577 Ibaraki, Japan
{hani,jiro}@iplab.cs.tsukuba.ac.jp

Abstract. In this paper, we propose a Two Handed Interactive Menu as an evaluation of asymmetric bimanual gestures. The menu is split into two parts, one for each hand. The actions are started with the non-dominant hand and continued with the dominant one. Handedness is taken into consideration, and a different interface is generated depending on the handedness. The results of our experiments show that two hands are more efficient than one; however the handedness itself did not affect the results in a significant way. We also introduce the Three Fingers Click, a selection mechanism that explores the possibility of using a depth-sensing camera to create a reliable clicking mechanism. Though difficult to maintain, our Three Fingers Clicking gesture is shown in the experiments to be reliable and efficient.

Keywords: bimanual gestures, depth-based click.

1 Introduction

Hand gestures have been investigated in Human Computer Interface, and bimanual gestures have been gaining popularity [1], [2], [6], [7]. Lévesque et al. have summed up in their research that two hands can perform better than one on a given task [13]. According to Guiard [3], bimanual gestures are classified into two parts: symmetric, where both hands are playing the same role (e.g. rope skipping) and asymmetric, where each hand is playing a different role (e.g. playing the violin).

In traditional desktop User Interface, menus are often used. Menus represent a structured way for displaying several options to the user. The advantage of menus is that, even though they can hold numerous options, they do so in a way that does not clutter the visualization surface. Several menu arrangements exist (dropdown menu, pie-menu, marked menu …), each with its own idiosyncrasies.

To the best of our knowledge, creating a menu system that is optimized not only for hand gestures, but for bimanual gestures as well, has not been attempted before. Therefore, we present our interactive menu as an approach of applying asymmetrical bimanual gestures in User Interface design.

S. Yamamoto (Ed.): HIMI 2014, Part I, LNCS 8521, pp. 187–198, 2014.

Gesture data fetching has been classified into 2 main categories: glove-based and vision-based [9]. Yang et al. state that because the glove-based method uses gloves and extra sensors, those extra sensors make it easy and accurate to collect gestures data, when compared with vision-based techniques [9]. However, using a vision-based technique allows the gestures to be recognized in an untethered way, thus freeing the user from donning any special hardware or apparatus to enable him to interact with the system which gives rise to a more natural interaction [10]. For this reason, we have opted for the vision-based approach.

In our study, we first try to find the best bimanual interaction method, which can convey instructions to the menu system. Then we propose a depth-based clicking method, as a way of allowing the user to select a given command. Finally, we put together a prototype, and we conduct a series of experiments to determine the feasibility and the performance of the proposed system.

2 Related Work

Among the earliest contributions to asymmetric bimanual gestures research is Guiard's work [3], which states that human bimanual interaction is asymmetrical; while both hands work together, the dominant and the non-dominant hands are doing different gestures. Guiard created a model for bimanual interaction, known as the "Kinematic Chain Model". Hinckley et al. have argued that, with appropriate design, two hands are not only faster than one hand, but they can also provide the user with additional information that a single hand alone cannot [1]. It has also been shown that users were able to perform complex commands in a natural way using mixed hands gestures [9]. Wagner et al. have also shown that bimanual interaction outperforms unimanual interactions [7].

However, not all bimanual interfaces are better than unimanual ones; in given situations, one-handed manipulation proved better than its bimanual counterpart [8], [11]. Chen et al. [14] have also shown that under certain circumstances, one-handed techniques were faster than two-handed techniques.

Applying bimanual interactions on menus has already been approached from different perspectives. In the bimanual marking menu [15], the marking is performed by the non-dominant hand. This has been confirmed as a very efficient bimanual technique [14]. In Guimbretière et al.'s study, to activate the menu system, the user performs a pinch gesture with his non-dominant hand, moves his hand, and finally releases the pinch to finish marking [16]. This design leaves the non-dominant hand free to participate in other gesture-based activities.

Typically, issuing a command (or choosing a menu item) should be performed with some kind of selection mechanism. One way is to release a previously executed pinch to perform a marking [16]; another approach would be to use the primary hand to point out to an item, and using a selection gesture performed by the non-dominant hand to select that item [13]. A touchscreen click simulating gesture can also be implemented with a depth camera as shown by Wilson [17].

3 Design Principles for Menu Interactions

In this section, we describe the interactions that we have designed for the Two Handed Menu. Guiard states that the non-dominant hand performs the coarse movements, whereas the dominant hand performs the fine and precise movements [3]; this formed the basis of our approach to creating interaction techniques.

Bimanual tasks give rise to a better performance if the action of the dominant hand depends on the action of the non-dominant hand [11]; the non-dominant hand executes the commands that require less precision, while the more precise actions are performed by the dominant hand. Applying this to a menu system, in a general way, stipulates that the non-dominant hand selects a sub-menu from the main menu; the dominant hand then selects the desired command from a sub-menu.

To make the system more complete, the following gestures will be used: “Show menu” which displays the menu on screen, “Go up” which allows the user to go to the previous submenu, and “Hide menu” which exits the menu without issuing any command. A selection gesture will also be used to allow the user to select a menu item. “Show menu”, “Go up” and “Hide menu” do not require precision, and thus can be assigned to the non-dominant hand.

This gives rise to a conflicting set of commands, such as “Show menu” and selecting a submenu, both being performed by the non-dominant hand. While it was shown that bimanual interfaces perform better than unimanual interfaces [2], [5], [6], [7] bimanual interfaces can induce a decreased performance if the interaction techniques are poorly designed [8], [11], [14].

As a first step, we have decided to find out which is the better interaction technique for commands that involve repeated use of the same hand.

3.1 Experiment 1 – Determining the Sequence of Interactions

As highlighted previously, some interactions cannot be separated into a sequence of “non-dominant hand, dominant hand” actions, rather some repetition with the same hand needs to be used at a given point. In this experiment, we aim at finding the better sequence when repetition is required.

The experiment consists of displaying a circular target at random positions. A small, hand-shaped cursor designates the current position of the user’s hand. The participant has 30 seconds to hit as many targets as possible. When a target is hit, a new target appears in a different position. Three variations of this experiment have been conducted:

1. One hand: the user uses only his dominant hand to hit the target.
2. Two hands – sequential: the target position is random, but it appears alternately on either side of the screen. The user has to hit the target with the hand corresponding to its relative side (left hand for the left side, right hand for the right side).
3. Two hands – random: the targets appear in a total random way. In this case too, the user uses the corresponding hand to hit the target.

To implement the experiment, we used a SoftKinetic DS325 depth sensing camera, which uses Time-of-Flight technology [18]. The camera was placed on top of a 23” monitor with Full HD resolution, facing the user, tilted down approximately 10 degrees. A prototype has been implemented on a computer equipped with an Intel Core i5 3.2 GHz CPU. The SDK of SoftKinetic has been used to detect hand tip positions. Onscreen rendering has been implemented in OpenGL. The entire prototype was written in C++. To evaluate our system, 9 participants (5 males, 4 females) aged between 23 and 30 were recruited; 6 among them are computer scientists/engineers. 8 of them are highly familiar with computers. 2 of them are left-handed.

3.2 Results

The results show a significant increase in performance when using two-hand gestures compared against using one hand only. In their study, Chen et al. found that, in given cases, using two hands sequentially was slower than using just one hand [14]. In our experiment, the results came contrary to that, showing that using two hands sequentially was the faster interaction: Fig 1.a. shows that the number of hits in 30 seconds is greatest for “Two-handed sequential” (42.67 with a standard deviation of 5.07 for n=9). Another result that was generated by this experiment is that the interaction slows down when consecutive actions had to be repeated by the non-dominant hand: Fig 1.b indicates that the average difference of time between each hit, as well as the maximum time difference between hits, are smallest for “Two-handed sequential” (0.74 and 2.46, with a standard deviation of 0.39 for n=9)

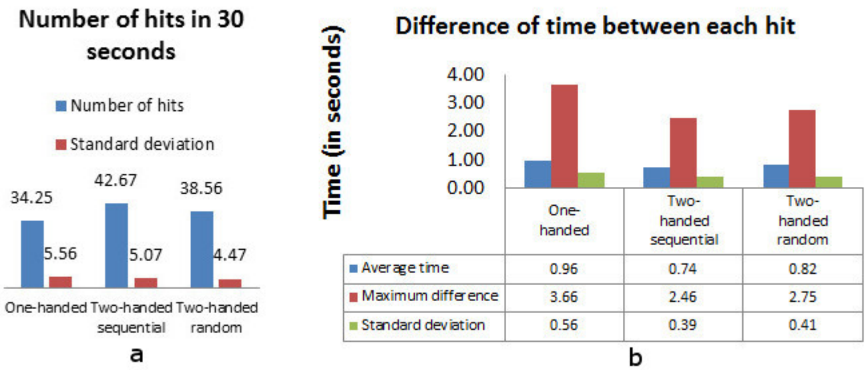


Fig. 1. Results of the experiment showing the number of hits in 30 seconds (a) and the difference of time between hits (b), for each of the three experiments variations

3.3 Interaction Design

Taking the results of the previous experiments into consideration, we have thus created the following rules for interacting with the system:

- The non-dominant hand interacts with the main menu, while the dominant hand interacts with all the submenus (a given submenu can lead to another submenu).
- "Go up" and "Hide menu" gestures do not require any precision, so they can be assigned to the non-dominant hand.
- While the "Show menu" gesture does not need any precision, it has been assigned to the dominant hand to avoid the scenario of showing the main menu and interacting with it using the non-dominant hand, conforming with the results of our experiment.

4 Three Fingers Clicking Gesture

To be able to instruct the system about a command (that is, selecting a menu item), some kind of interaction is required. In this section, we introduce the "Three fingers clicking gesture", a novel selection mechanism approach.

Some previous work consists of using the non-dominant hand to initiate this command, such as "index pointing, thumb up" gesture [13], or releasing a previously executed pinch [16]. The disadvantage of these models is that they do not rely on an intuitive way to perform the operation. An intuitive approach is to imitate the finger clicking gesture, widely used on touch displays. In some situations, a calibration of the environment is needed; Wilson calibrated the system by using a depth threshold determined from a histogram over several hundred frames of a motionless scene [17]. An ad-hoc approach that does not use calibration exploits a flood filling technique to detect whether a finger has clicked a surface [20]. These two techniques detect a physical contact with a surface. A mid-air clicking gesture is proposed in OpenNI [19]: an "L" shape is created by extending the index and thumb fingers to signal the start of the clicking gesture; then the gesture itself is performed by pushing the entire hand away from the user's body.

In our approach, we assume that a finger clicking event occurs when the index finger passes beyond a given threshold. To define this threshold, we detected the X, Y and Z coordinates of the thumb, index, and middle fingers (noted T, I and M respectively). We also detected the same information about the hand palm's center (noted P). We have defined the plane created by the points [P, T, M]. The angle θ between this plane and the vector [PI] is then computed (Fig. 2). A threshold of 12 degrees was selected empirically. The test is performed within one frame.

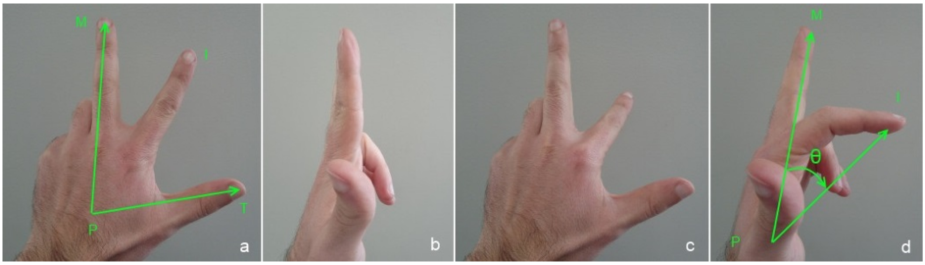


Fig. 2. 1-a and 1-b depict an unclicked state. 1-c and 1-d depict a clicked state

To perform the gesture, we assume that only the thumb, index and middle fingers are held open; the index is then bent forward to initiate a click. The advantage of this model is that no calibration is required. Another advantage is that the reference against which the threshold is tested is always moving along with the index finger; this gives the user the freedom of moving his hand in mid-air prior to performing the gesture. And since our approach does not rely on analyzing previous frames and comparing the position of the index against them, the result of the gesture is instantaneous. Because of how our prototype was designed, we were not able to accurately measure the detection speed. We plan on conducting a deeper evaluation in future work.

4.1 Experiment 2 – Determining the Accuracy of the Three Fingers Click

For this experiment, we have used the same setup as Experiment 1; the only exception being that rendering was done using the Allegro library. The same 9 participants that took part in Experiment 1 were also recruited. Handedness was not taken into consideration in this experiment, and the participants were instructed to hold open the thumb, index and middle fingers of their preferred hand, and perform 20 clicks with their index finger. A human observer counted the total number of clicks performed, in order to spot false detections.

4.2 Results

The average number of click attempts performed by the users to complete 20 clicks was 22.33. This indicates a success rate of 89.56%. 3 participants had a 100% success rate.

Using 3D coordinates allowed the gesture to be detected regardless of the hand's position or rotation. The gestures were detected even if the wrist was slightly rotated inwards, the palm was slightly pointing downward, or even if the hand was moving. This is due to the fact that the index finger and the reference all move as a single group. While in this experiment, the camera is facing the user, this technique is also applicable even if the camera is behind the user's palm and facing away from the user; the clicks were also being detected in the latter position. We suppose that our design can also be applied to tabletop setup, with a depth sensing camera pointing downwards.

Since the gesture relies on the detection of three fingers, this can be a detection/accuracy limitation. When a given hand is in its own half of the camera space, the three fingers were easily detected. However, when the hand moved into its opposite half of the camera space, the finger detection failed, even if the hand was still in the camera's field of view. This is due to the fact that when the hand crosses into the opposite space, the thumb and the index are occluded by the middle finger, and the camera fails to keep track of them (Fig. 3). Another limitation is the gesture itself: 6 participants reported that keeping the three fingers held open stressed their forearm's muscles quickly, and found some difficulty in maintaining the gesture.

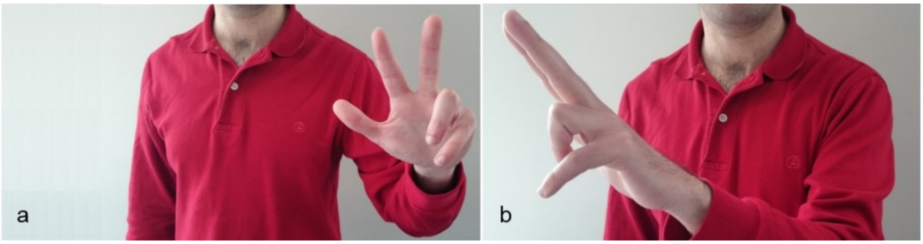


Fig. 3. The fingers of the left hand in the left half of the camera space are easily detected (a), however, detection fails when the left hand moves to the right half of the camera space (b)

5 Two Handed Menu

In this section, we present a prototype for a “Two Handed Menu”. For this intent, we would like to create a new menu system that is optimized, not only for hand gestures, but for two hands as well.

Since in real-world human interaction, asymmetric bimanual motions start with the non-dominant hand, and are then followed by the dominant hand [3], we will create a menu system that is split into two parts, thus allowing the user to interact with it using both his hands. The menu system will be handedness-free, meaning that it will take into consideration whether the user is right or left-handed, and dynamically generate the appropriate user interface depending on the hand preference. We believe that having data from a group of mixed handedness participants will allow us to better evaluate the system.

Basing our menu on the traditional desktop toolbar menu, we have created a “Main Menu” containing the following items: “File”, “Edit”, “View”, and “Help”. Any menu spawning from the selection of one of those 4 items is designated as “Submenu”. A submenu can spawn its own submenu. A hand-shaped cursor indicates the user’s current hand tip position. The user interacts with the main menu using his non-dominant hand, while he uses his dominant hand to interact with all of the submenus. For this reason, the main menu is displayed on the non-dominant hand’s side, whereas the submenus are displayed on the dominant hand’s side. The user moves his arms up and down to be able to hover above the menu items, and then selects an item using the Three Fingers Click described in Section 4.

Figure 4 depicts a right-handed layout of the menu. In figure 4.a, the main menu (File, Edit, View, Help) is rendered. When the user selects “Edit” with his left hand, the Edit sub menu (Undo, Redo, Find and Replace, Select All) is then rendered on the right side of the display as shown in Figure 4.b. Upon clicking “Find and Replace” with the right hand, a new sub menu is then rendered in Figure 3.c (Quick Find, Quick Replace). In this last case, if the user performs a “Go up” gesture, he will go back to 4.b, and from there another “Go up” gesture will take him back to 4.a.

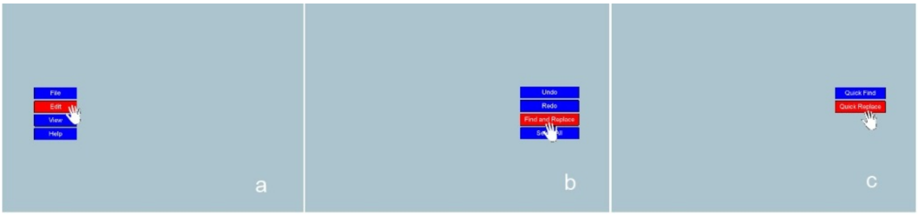


Fig. 4. Right-handed menu

Figure 5 shows a similar example, but rendered for a left handed user. In this case, the user interacts with the main menu using his right hand (Figure 5.a), then interacts with any other submenu using his left hand (Figure 5.b).

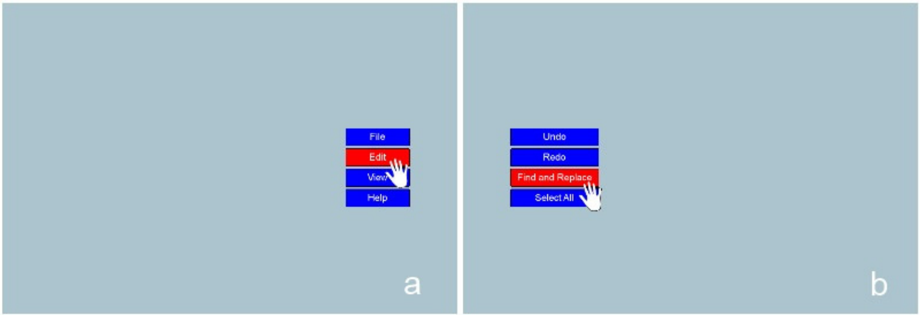
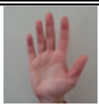
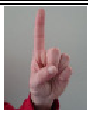
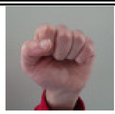


Fig. 5. Left-handed menu

Drawing conclusions from section 3, we have created a set of gestures to interact with the menu system. Table 1 shows the gestures that have been used in this prototype.

Table 1. Hand gestures used in the prototype

			
Show menu	Go up	Hide menu	Point / Click

An open palm is used to display the menu. Pointing up with the index instructs the system to go up one level in the menu. A clenched fist hides the menu, in case the user wants to exit the menu without issuing a command. The above mentioned "Show menu", "Go up" and "Hide menu" gestures are static gestures. As described in Section 3 as well, "Show menu" will be assigned to the dominant hand, whereas "Go up" and "Hide menu" will be assigned to the non-dominant hand. "Point/Click" will be used by both hands.

5.1 Experiment 3 – Testing the Two Handed Menu Prototype

For this experiment, we have used the same setup as Experiment 2. The same 9 participants that were involved in Experiments 1 and 2 were also recruited. They were asked to perform the scenario described in Table 2:

Table 2. Gestures to be performed by either hand. A blank space indicates that the hand in the same row does not perform any action.

Dominant	Show		Close	Show		Find and Replace		Undo		Exit
Non dominant		File			Edit		Go up		Show	

First, the participants were asked to perform the above scenario using their usual handedness. Next, we switched the layout of the menu, and asked them to perform the same scenario; all the commands now have switched handedness as well.

The participants were also asked to perform the same scenario using their preferred hand only.

A final test was performed to evaluate the ergonomics of the hand motions. Tomita et al. have proposed a slanted menu as a more ergonomic approach [4]. In our system, the menu "morphed" with the hand and positioned itself around the participants' hand tip. Instead of using an up/down motion to interact with the menu, the participants were able to use a waving motion from top left (or top right) to the bottom center of the display (Figure 6).

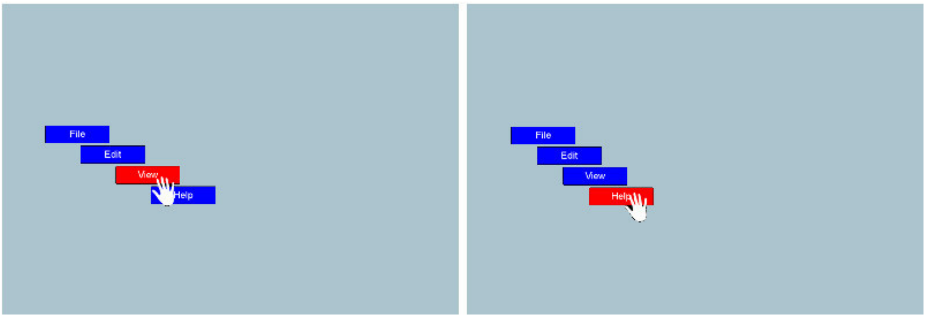


Fig. 6. Morphing menu that follows the user's hand

5.2 Results

Before presenting the results of this experiment, we duly note that no participant was able to complete the last scenario, which uses one hand only; this is mainly due to the limitation described in section 4.2. Even though the users attempted to circumvent the limitation by twisting their wrists in a way so that the fingers can be detected, the unease resulting from the strain put on the wrist and shoulder made it impossible for them to continue the scenario. Thus, no comparison between one and two handed operations was possible.

We measured the time it took the participants to complete the given scenario. When the participants used their usual handedness, the average time to complete the scenario was 31.16 seconds (standard deviation: 6.58), whereas the average time to complete the reversed handedness scenario was 35.47 seconds (standard deviation: 15.77. The numbers were close due to the fact that the participants were moving their hands in similar fashion for either case, as well as the fact that the menu input did not require extreme precision (thus conforming to the "easy task" in [12]).

6 Evaluation

After completing the experiments, the participants were handed out the following questionnaire, to which they could respond on a five point Likert scale (-2 = Strongly Disagree, 2 = Strongly Agree):

1. Is using two hands easier than using one hand?
2. Is it easy to maintain the Three Fingers gesture?
3. Does the Clicking Gesture simulate a mouse click?
4. Is using an interface tailored to handedness easier?
5. Does using only one hand strain the shoulder/wrist?
6. Is using a Morphing Menu more natural than a traditional layout?

Most participants agreed that using two hands was easier than one hand with an average of 1.22 (standard deviation: 1.09), and most found that the Three Fingers Gesture was difficult to maintain (-0.88, 1.05). Everyone agreed that the clicking gesture simulates a mouse click (1.44, 0.52). There were mixed results regarding tailoring the interface to the handedness (0.44, 0.88). One of the left-handed participants, who uses the mouse with her right hand, gave a negative answer regarding that question. Everyone agreed that using only one hand was uncomfortable (1.77, 0.44). There were mixed results for the morphing menu as well (0.55, 1.01); the users who liked this variation stated that it felt more natural to wave the hand rather than go up and down, and it was less tiring because they were able to rest their elbows on the desk or on their lap.

7 Conclusion and Future Work

In this paper, we have presented a menu designed with a bimanual interface. The goal was to create a new approach on User Interfaces, by using the hands asymmetrically to control a menu. Our experiments showed that two hands were faster than one, but handedness itself did not affect the performance in a significant way in this specific prototype.

We have also introduced the Three Fingers Click, a novel and reliable clicking mechanism that does not need calibration.

Some design considerations were found as well, which could serve as a reference for future interface designs, especially when using a setup like ours: if fingers are to be used in an interface, the hand should move in its own half of the camera space, due to the limitations of the wrist and shoulder anatomy; thus, an interface using fingers should be designed for either one hand / one half of the camera space, or two hands across the entire camera space.

In our future work, we would like to explore in more details the depth selection mechanism that we have introduced, especially that our current prototype was not designed in a way to allow a proper quantitative assessment of its performance. We feel that our approach could serve as a base for some interesting depth based selection mechanisms. To make the mechanism easier to use, we would also like to extend it to 5 fingers in future designs.

References

1. Hinckley, K., Pausch, R., Proffitt, D., Kassell, N.: Two-handed virtual manipulation. ACM Transactions on Computer-Human Interaction 5, 260–302 (1998)
2. Veit, M., Capobianco, A., Bechmann, D.: Consequence of two-handed manipulation on speed, precision and perception on spatial input task in 3D modelling applications. Universal Comp. Science 14, 3174–3187 (2008)
3. Guiard, Y.: Asymmetric division of labor in human skilled bimanual action: The kinematic chain as a model. Motor Behavior 19, 486–517 (1987)
4. Tomita, A., Kambara, K., Siio, I.: Slant menu: novel GUI widget with ergonomic design. In: Proceedings of CHI 2012 Extended Abstracts on Human Factors in Computing Systems, pp. 2051–2056 (2012)
5. Song, P., Boon Goh, W., Hutama, W., Fu, C., Liu, X.: A Handle Bar Metaphor for Virtual Object Manipulation with Mid-Air Interaction. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp. 1297–1306 (2012)
6. Yang, R., Strozzi, A., Lau, A., Lutteroth, C., Chan, Y., Delmas, P.: Bimanual natural user interaction for 3D modelling application using stereo computer vision. In: Proceedings of the 13th International Conference of the NZ Chapter of the ACM's Special Interest Group on Human-Computer Interaction, pp. 44–51 (2012)
7. Wagner, J., Huot, S., Mackay, W.: BiTouch and BiPad: designing bimanual interaction for hand-held tablets. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp. 2317–2326 (2012)
8. Guimbretière, F., Martin, A., Winograd, T.: Benefits of merging command selection and direct manipulation. ACM Transactions on Computer-Human Interaction 12, 460–476 (2005)
9. Yang, Z., Li, Y., Zheng, Y., Chen, W., Zheng, X.: An Interaction System Using Mixed Hand Gestures. In: Proceedings of the 10th Asia Pacific Conference on Computer Human Interaction, pp. 125–132 (2012)
10. Boussemart, Y., Rioux, F., Rudzicz, F., Wozniewski, M., Cooperstock, J.: A framework for 3D visualisation and manipulation in an immersive space using an untethered bimanual gestural interface. In: Proceedings of the ACM Symposium on Virtual Reality Software and Technology, pp. 162–165 (2004)
11. Kabbash, P., Buxton, W., Sellen, A.: Two-Handed Input in a Compound Task. In: Proceedings of ACM CHI Conference, pp. 417–423 (1994)

12. Hinckley, K., Pausch, R., Proffitt, D., Patten, J., Kassell, N.: Cooperative Bimanual Action. In: Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems, pp. 27–34 (1997)
13. Lévesque, J.C., Laurendeau, D., Mokhtari, M.: Bimanual Gestural Interface for Immersive Virtual Environments. In: Proceedings of the IEEE Virtual Reality Conference, pp. 223–224 (2011)
14. Chen, N., Guimbretière, F., Löckenhoff, C.: Relative role of merging and two-handed operation on command selection speed. International Journal of Human-Computer Studies 66(10), 729–740 (2008)
15. Odell, D., Davis, R., Smith, A., Wright, P.: Toolglasses, marking menus, and hotkeys: a comparison of one and two-handed command selection techniques. In: Proceedings of Graphics Interface, pp. 17–24 (2004)
16. Guimbretière, F., Nguyen, C.: Bimanual marking menu for near surface interactions. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp. 825–828 (2012)
17. Wilson, A.: Using a depth camera as a touch sensor. In: ACM International Conference on Interactive Tabletops and Surfaces, pp. 69–72 (2010)
18. DepthSense 325, `http://www.softkinetic.com/Store/tabid/579/ProductID/6/language/en-US/Default.aspx`
19. OpenNI gestures, `http://www.openni.org/files/igesture3d/`
20. Harrison, C., Benko, H., Wilson, A.: Omnitouch: wearable multitouch interaction everywhere. In: Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology, pp. 441–450 (2011)