Joint Virtual Reality Conference of EGVE - ICAT - EuroVR (2009)
M. Hirose, D. Schmalstieg, C. A. Wingrave, and K. Nishimura (Editors)

Designing 3D Selection Techniques Using Ballistic and Corrective Movements

Lei Liu[†] and Robert van Liere[‡]

Centrum Wiskunde & Informatica, the Netherlands

Abstract

The two-component model is a human movement model in which an aimed movement is broken into a voluntary ballistic movement followed by a corrective movement. Recently, experimental evidence has shown that 3D aimed movements in virtual environments can be modeled using the two-component model. In this paper, we use the two-component model for designing 3D interaction techniques which aim at facilitating pointing tasks in virtual reality. This is achieved by parsing the 3D aimed movement in real time into the ballistic and corrective phases, and reducing the index of difficulty of the task during the corrective phase. We implemented two pointing techniques. The 'AutoWidth' technique increases the target width during the corrective phase and the 'AutoDistance' technique decreases the distance to the target at the end of ballistic phase. We experimentally demonstrated the benefit of these techniques by comparing them with freehand aimed movements. It was shown that both 'AutoWidth' and 'AutoDistance' techniques exhibit significant improvement on target acquisition time.

Categories and Subject Descriptors (according to ACM CCS): I.3.6 [Computer Graphics]: Methodology and Techniques—Interaction techniques I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Virtual reality I.3.6 [Computer Graphics]: Methodology and Techniques—Ergonomics

1. Introduction

The most primitive interaction tasks in virtual environments are direct 3D selection and pointing tasks. The efficiency and performance of 3D selection and pointing tasks have been studied extensively, such as [WS91], [MI01] and [GB04], etc. These studies have examined the difficulties associated with pointing and the factors that influence the performance of pointing. In parallel, human computer interface researchers have designed and developed interaction techniques to improve the performance and efficiency of pointing tasks. Some interesting 2D examples of such techniques are 'semantic pointing' [BGBL04] which adapts the control-display ratio depending on the distance to a target, 'Drag-and-pop' and 'drag-and-pick' [BCR*03] that remotely drags potential targets towards current cursor location, 'Area cursor' [WWBH97] which has a larger than regular activation area to seize more opportunities of stopping cursor on a target, etc. Each of the examples relies on the distance between the cursor and the target to adjust some aspects in the visual or motor space.

In this paper, we take a different approach in designing interaction techniques for 3D selection tasks. We use the two-component model for aimed movements as a model to determine how the user approaches a target. The two-component model is a human movement model in which an aimed movement is broken into a voluntary ballistic movement followed by a corrective movement [Woo99]. It has been shown that the velocity profiles of these phases are very different. Figure 1 is an example of a typical velocity profile. Two-component model has been studied in real world settings, but recently we have shown that it can be applied to 3D pointing and selection tasks in virtual reality as well [LvLNM09].

A different widely used model in HCI is Fitts' law [Fit54], where the movement time of a pointing task is modeled as

[†] Lei.Liu@cwi.nl

[‡] Robert.van.Liere@cwi.nl

DOI: 10.2312/EGVE/JVRC09/001-008

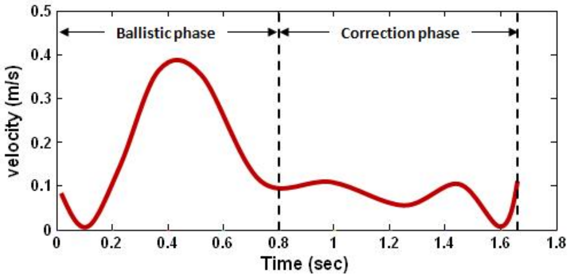

Figure 1: *The ballistic phase and corrective phase of a typical 3D aimed movement, shown in velocity profile.*

a function of the distance to the target and the size of the target. Fitts' law has been formulated in different ways. One common formulation is:

$$T = a + b\,\log_2\left(\frac{D}{W} + 1\right) \qquad (1)$$

where a and b are constants that can be determined experimentally. D is the distance to the target, while W is the target width. $\log_2(D/W + 1)$ is often called ID, indicating the index of difficulty of a pointing task under certain environment.

Our aim is to design 3D selection techniques by combining the two-component model for aimed movements with Fitts' law. The general idea is to parse the aimed movement in real-time into ballistic and corrective movements, and reduce the index of difficulty of the task during the corrective phase. Similar interaction techniques have been proposed in 2D desktop environment, but, to our knowledge, this idea is new for spatial interaction. We have implemented two selection techniques. The 'AutoDistance' technique has been designed to decrease the distance to the target at the end of ballistic phase and automatically snap the cursor onto the target. The 'AutoWidth' technique increases the target width during the corrective phase. From our experimental results, both techniques have significant improvements on reducing the movement time of 3D aimed movements, when compared to freehand interaction scenario.

The main contributions of the paper are:

- the design and development of 3D interactive selection techniques by combining the two-component model and Fitts' law;
- the real-time 3D movement parsing criteria;
- the two implemented interaction techniques, 'AutoWidth' and 'AutoDistance', based on the proposed idea;
- and the experimental evaluation of the feasibility and effectiveness of the techniques.

2. Related work

Balakrishnan [Bal04] has studied a similar approach for enhancing 2D pointing tasks. He proposed to decrease the distance to target in the ballistic phase, and to increase target width in the corrective phase. This work adopted Meyer's stochastic optimized sub-movement model [MAK*88] to take different movement phases into account. In motor space, Worden, et al [WWBH97] developed techniques for which the control-display ratio remained high during the initial ballistic phase. Further, the control-display ratio was reduced during final corrective phase where cursor velocity is relatively low. These techniques were designed for single isolated 2D target acquisition. If interaction techniques are designed for multiple 3D targets, the difference between both 2D vs. 3D aimed movement characteristics and single vs. multiple targets should be considered carefully.

One example to improve multiple-3D-target selection in virtual environments is through dynamically scaling targets and forced disocclusion [AA08]. These 3D pointing techniques were designed especially for ray-casting, independent of direct manipulation. On the contrary, the Go-Go immersive interaction technique [IP96] manipulated the 3D object with a virtual hand which had a linear mapping to user's real hand within a certain distance, but a non-linear mapping to make the virtual hand "grow" otherwise. Go-Go is equivalent to reduce the distance between the virtual hand and the target when reaching to a remote target, but it can only improve the efficiency in terms of remote target acquisition. Frees et al presented 'PRISM' [FKK07] for directly manipulating 3D objects in immersive environment. Depending on the hand speed of the user, it dynamically adjusts the C/D ratio in such a manner that hand movement can be scaled when accuracy and precision is needed, while it is free of any artificial constraints when moving rapidly. However, PRISM only takes the hand speed into account and the implementation relies strongly upon the selection of the thresholds 'MinS', 'SC' and 'MaxS' which determine the C/D ratio. In this paper, we suggest to further make use of the speed, acceleration (first derivative of speed in terms of time) and jerk (second derivative) of the hand movement to explicitly break movement into distinct phases in real time and apply intercalation techniques in some of them.

In previous work, we have compared 3D aimed movements in the real world with aimed movements in virtual environments [LvLNM09]. We have shown that velocity profiles of the ballistic phases are very similar. However, the time taken in the corrective phase is significantly longer in the virtual environment than in the real world. This may lead to a different ballistic phase time / corrective phase time ratio compared to that of 2D aimed movements. Therefore, we should concentrate on the corrective phase which involves large amounts of time while moving only a relatively small distance.

3. Designing selection techniques using two-component model

Fitts' law models the movement time of an aimed movement as a function of the index of difficulty (*ID*), which depends on the distance to target (*D*) and target width (*W*) parameters. Accordingly, to reduce the movement time, an interaction technique can decrease the distance to target, increase the target width, or even change both of parameters simultaneously. The two-component model, describes an aimed movement into a ballistic phase and a corrective phase. The ballistic phase usually covers bulk of the distance to target with high velocities. The corrective phase, although traversing only in the vicinity of the target, takes a lot of time due to the low velocities and small adjustments. If the two movement phases could be distinguished in real time, we can apply different strategies to each movement phases so as to reduce the movement time in each phase separately.

	Ballistic	Corrective	Total movement
D↓	①	②	③
W↑	④	⑤	⑥
D↓ & W↑	⑦	⑧	⑨

Table 1: *The general idea of designing interaction techniques for 3D pointing and selection tasks. D↓: decreasing the distance to target; W↑: increasing target width.*

Table 1 lists the nine possible strategies of reducing the movement time. For example, one strategy could involve only decreasing the distance to the target in the corrective phase (②). Strategies can also be combined. For example, combining ① + ⑤ would decrease the distance to target in ballistic phase while increase the target width in corrective phase.

To decrease *D* in visual space, the cursor can be automatically moved towards the target as in 'snap-dragging' [Bie88]. To increase *W*, we can either expand target width visually, as in Apple's Mac OS X "dock", or expand the cursor width visually, as 2D 'Area cursor' [WWBH97]. In addition to adjusting the visual space, the motor space can also be altered. For example, the motor space can be scaled by adjusting control-display ratio during the ballistic or corrective phases. Therefore, table 1 provides considerable possibilities to design interaction techniques for facilitating 3D aimed movements.

3.1. Real-time movement parsing

Meyer, et al proposed a number of 1D movement parsing criteria in the stochastic optimized sub-movement model [MAK*88]. The idea was to divide a 1D aimed movement into 3 basic types of sub-movements and assemble the sub-movements into phases. In previous work, we have extended Meyer's criteria to 3D movements, [LvLNM09]. The implementation of our criteria parsed recorded movement trajectories as a post-processing step. However, in this experiment, movement parsing needs to be done in real time while subjects are reaching the target. The absence of global overview for the complete movement makes it difficult to discriminate corrective phase from ballistic phase. For instance, the corrective phase can only start after the global peak of a velocity profile has detected. But in real time, it is not possible to distinguish between the global peak and a local peak of the velocity.

We introduce a procedure which can parse 3D movement in real time. The entire procedure involves 5 steps: data preprocessing, global peak detection, sub-movement detection, end of ballistic phase detection and target prediction.

During *data preprocessing*, a velocity profile is constructed after a position sample has been received from the input device tracker; e.g. every 1/120 sec. The velocity profile is smoothed by taking the average of velocity values every 10 samplings. We also compute the acceleration and jerk of the smoothed velocity.

The *global peak* of a velocity profile is detected if all the following three conditions are met:

- A zero-crossing of acceleration from positive to negative is reached;
- The velocity is greater than a threshold *a*;
- The time spent is longer than a threshold *b*.

The thresholds *a* and *b* ensure small local peaks in the velocity profile are not considered as the global peak. They are derived from the pre-experiment where *a* and *b* are the minimum values to become a peak velocity.

Part of a movement is defined as a *sub-movement* when any of the three conditions is met at the end of the sub-movement:

- The velocity is smaller than a threshold *c* (type 1);
- A zero-crossing of acceleration from negative to positive is reached (type 2);
- A zero-crossing of jerk from positive to negative is reached (type 3).

The criteria above resemble Meyer's 1D movement parsing criteria, except that type 1 sub-movement was defined as a zero-crossing of velocity from positive to negative in Meyer's criteria. Because, in 3D space, we can hardly obtain a zero velocity at any time during the movement due to the jitters from the human motor system and the magnetic tracking system. Threshold *c* is the maximum value which can be deemed immobility in the pre-experiment.

The *end of the ballistic phase* is defined as the moment all the following conditions are satisfied:

- The global peak has been observed;
- A type 1, 2 or 3 sub-movement is detected;

- The current position is within a reasonable distance d from the target.

Threshold d is determined as such a distance from the target where most of the ballistic phases end in the pre-experiment.

After having reported the end of ballistic phase, we introduce the 'nearest neighbor' as the *target prediction algorithm*, since we assume the cursor has already entered the vicinity of the intended target at the moment. Therefore, there is a little chance to pick a wrong target. It happens when the subject intend to reach a wrong target or has a dramatic change in behavior while passing through an unexpected target. But in practice this rarely occurs.

3.2. Interaction techniques

Two interaction techniques have been implemented.

AutoWidth is the interaction technique which expands the 3D target to a fixed size during the corrective phase of the aimed movement (⑤ in table 1). AutoWidth takes effect immediately after the moment that the parsing algorithm reports the end of ballistic phase and a target has been predicted. In the experiment, the expanded target size was set to be twice as large as the original size (see algorithm 1 for pseudo-code).

Similarly, AutoDistance is defined as the technique in which the cursor in the visual space is dragged toward the predicted 3D target and snaps to the center of the target immediately after the end of ballistic phase (② in table 1). The cursor was caught so firmly that it won't be released until the stylus in the motor space moves faster than a predefined threshold e. When snapped, the cursor has only 3 DOF, i.e. 3-axis rotation, and the translation is locked (see algorithm 1 for pseudo-code).

As described, the snap dragging only takes place in the visual space. In motor space, however, subjects feel nothing unusual, i.e. no haptic feedback. So visuo-proprioceptive conflicts are generated. Having trained for several trials, subjects were able to quickly adapt to it. Since snap dragging involves translating and translation lock during the movement, there tends to be a cumulative effect on the difference between the original tracked position and the translated tracked position, which can lead to a strong deviation of hand position from the center of the motor space. At the end of each trial, the cursor is translated back to the original place.

Freehand is the scenario where there is no aid provided during the pointing and selection tasks. But real-time movement parsing criteria have been applied to it as well with the aim of comparing it with AutoWidth and AutoDistance.

Algorithm 1 Interaction techniques

```
for MotionEvent of RenderWindowInteractor do
  if GlobalPeak==1 and (SubmovementType1==1 or SubmovementType2==1 or SubmovementType3==1) and DistCursorTarget<=$d$ and PredictTarget==TargetA then
    if AutoWidth==1 then
      WidthTargetA=2*WidthTargetA
    else if AutoDistance==1 and VelocityCursor<$e$ then
      PositionCursor = PositionTargetA
    end if
  end if
end for
```

4. Experiment

4.1. Apparatus

The experiment was performed under a desktop virtual environment, including a PC equipped with high end GPU, the Polhemus FASTRAK used to sample a 6 DOF stylus tracker at 120Hz, a Samsung HL67A750 67-inch 3D-capable LED DLP HDTV, a pair of Crystal Eyes stereoscopic LCD glasses and an ultrasound Logitech 3D head tracker working at 60Hz. During the experiment, the resolution of monitor was set to be 1920*1080 at 120Hz. The overall end-to-end latency of the virtual system was measured to be around 80ms using the method proposed by Steed [Ste08].

4.2. Subjects

There were 11 males and 7 females, aged from 28 to 45 years (average 32.1), voluntarily participating in the experiment. Half of them were 3D-VR-naive users, 6 had experience working with VR and 3 were well-skilled-VR users. They were all right-handed. 6 of them, half females and half VR-naive users, were invited to do the pre-experiment with the purpose of acquiring the proper thresholds (see section *movement parsing*). The remaining 12 subjects were instructed to perform the same experiment with thresholds obtained from pre-experiment.

4.3. Experiment setup

The experiment was performed in a non-collocated 1:1 sized condition (see figure 2). Subjects needed to wear a helmet onto which a head tracker was attached and stereo glasses while holding the stylus using their dominant hands. The focal point of the camera was set in such a way that the scene was coming out of the screen. The center of the visual space was 0.75m in front of the subjects when they were seated, while motor space was 0.3m from the subject, resulting in a distance of 0.45m between visual and motor space. The scene, resembling the ISO 9241 part 9 pointing task [Smi96], included a 0.4*0.4*0.28 sized box encapsulating 12 sphere targets and 1 sphere source, each of which was connected to a semi-transparent vertical column of the same size to the sphere on top of it. To enhance the depth perception, the floor of the box was covered by a virtual chessboard. Although we made sure that there were 3 targets in each of the quadrant of x-z plane, the positions of the targets were randomly generated. So, the distances between the source and targets were different from one target to another.

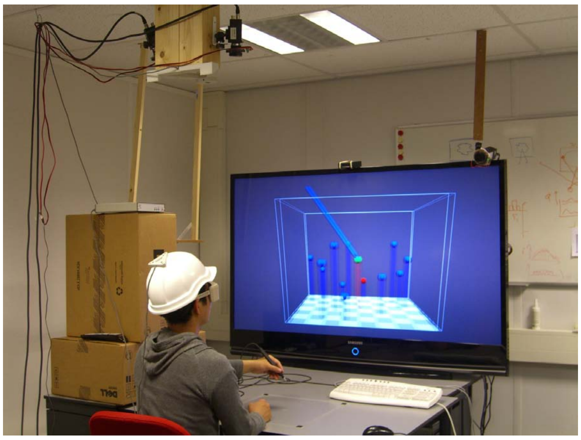

Figure 2: *Experiment setup*

The experiment was a multi-directional aimed movement task where subjects always started from the source and rapidly reached the indicated target. There was only one indicated target colored with red at the start of each trial and other targets remained in blue. The sphere, no matter a source or a target, turned to green at any time when the tip of the cursor was within it. Subjects were asked to start a trial by clicking the button of the stylus within the source. If subjects failed to do so, a warning message was shown, indicating that they had to start it again. If they succeeded, they noticed a darker background and brighter source and targets so that they were aware of the start of the goal-directed aiming. Meanwhile, the source and the indicated target became yellow, and other targets still remained in blue. Once subjects started, they needed to stop within the indicated target by clicking the button again. If they failed, nothing was noted and they must keep trying until successful. If they succeeded, the trial ended and the number of remaining trials was presented. Data were recorded only between a valid press within the source followed by a valid release within the indicated target, so subjects could have a rest and resume whenever they wanted between trials. They needed to repeat the above steps until the number of remaining trials became zero.

4.4. Procedure

The experiment was a repeated measures design with 60*3*18 (number of trials * number of blocks * number of subjects) times repeats, among which 60*3*6 were used for pre-experiment threshold acquisition and 60*3*12 for ANOVA analysis. Subjects reached one of the 12 targets 5 times randomly, constituting 60 trials. Trials were then grouped into 3 blocks, to which Freehand, AutoWidth and AutoDistance were applied, respectively. We gave trials in a block a random order which, however, was fixed for a subject's three blocks. Pre-experiment was performed before the actual experiment started and had the same procedure to the actual experiment, except that the thresholds mentioned in section *Real-time movement parsing* weren't included. To compensate the practice effect, either interference or learning effect, we adopted the incomplete repeated measures design [SZZ06] where 12 subjects were equally put into 6 groups. Subjects in different groups had to undergo all blocks, but were given in various orders. Before we collected the data, subjects were asked to practice an equal number of trials to the actual experiment using each of the 3 techniques and the order was the same with that of the corresponding actual experiment.

5. Results

All subjects confirmed that both interaction techniques are much more helpful and easier to control in acquiring the target than Freehand. 11 subjects out of 12 reported that AutoDistance is more helpful than AutoWidth and 1 reported the other way round.

5.1. Total movement time

Figure 3 shows the means of total movement time of Freehand, AutoWidth and AutoDistance among 12 subjects and the 95% confidence intervals correspondingly. Although fluctuating from subject to subject, it is clear that AutoDistance is the most efficient technique and Freehand, the least. According to the ANOVA results of the transformed data, the total movement time of AutoDistance (e.g. $M_{user1} = 1.2795$, $SE_{user1} = 0.0295$) is significantly different (e.g. $F_{user1}(1,118) = 31.73$, $p_{user1} = 1.2224e-7$) from that of the corresponding Freehand (e.g. $M_{user1} = 1.5846$, $SE_{user1} = 0.0566$) for each of the subject. Data also show significant differences between 12 subjects' AutoWidth and Freehand. Although AutoDistance always results in shorter duration than AutoWidth, only half subjects' data support that there is significant difference. User 5 and 12 are the slowest two among the 12, but we do notice a significant improvement on their total movement time of AutoDistance and AutoWidth averagely and user 5 is the one who has the greatest progress, 1.3522s and 0.9498s respectively, compared to its Freehand.

5.2. Ballistic phase time

As depicted in figure 4, there is no possibility to conclude which technique has a shorter ballistic phase. The null hypothesis "the means of 3 groups are all equal" can't be rejected at the 95% level of confidence by most of the data.

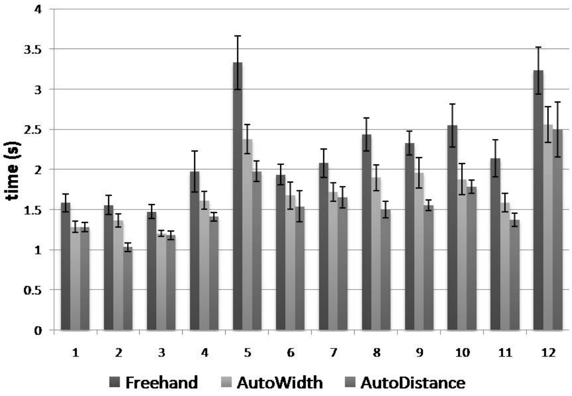

Figure 3: *total movement time: Freehand, AutoWidth vs AutoDistance*

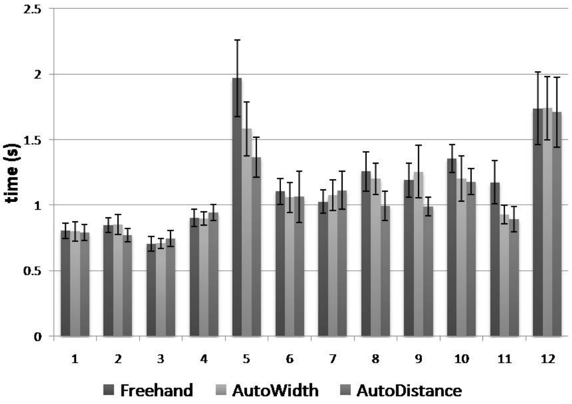

Figure 4: *ballistic phase time: Freehand, AutoWidth vs AutoDistance*

User 1, 3, 4, 6, 7, 12 exhibit thoroughly no difference (e.g. $F_{user1}(2,177) = 0.21$, $p_{user1} = 0.8127$) in the average of ballistic phase time among Freehand, AutoWidth and AutoDistance, as expected. Other users show a slight variation, especially user 5 and 11 whose ballistic phases of Freehand are significantly longer (e.g. $F_{user5}(2,177) = 8.49$, $p_{user5} = 0.0003$) than those of AutoWidth and AutoDistance. The fact is that both of them are completely naive users and have done the experiment in such an order that Freehand was followed by AutoWidth and then AutoDistance. The corresponding ballistic phase time is descending due to the fact either their ballistic phases were also affected by the interaction techniques designed to reduce the corrective phase time or they were still influenced by the learning effect. The former reason could be rejected by other users' performance. Therefore, it is clear, although required to practise before starting the experiment, some of the completely naive subjects still exhibited a lack of practice. Various trends are found from the rest of the data, but the differences between scenarios are very small.

5.3. corrective phase time

The trend of corrective phase time is similar to that of total movement time, except that the differences between Freehand and AutoWidth or AutoDistance are even greater. Both AutoWidth and AutoDistance are significantly different from their corresponding Freehand in terms of all users. Corrective phase time of AutoDistance is shorter than that of AutoWidth for most users, expect user 1 ($M_{AutoWdith} = 0.4868$ vs $M_{AutoDistance} = 0.4870$). The differences between AutoWidth and AutoDistance are, however, getting greater. There are 8 user's data showing significant differences (e.g. $F_{user2}(1,118) = 82.87$, $p_{user2} = 2.6645e-15$). The greatest difference between AutoWidth and Freehand is 0.6728s from user 12, and the least 0.2016s from user 2, while the greatest difference between AutoDistance and Freehand is 0.8643s also from user 12, and the least 0.2939s from user 1. Generally speaking, AutoDistance is far more helpful than AutoWidth.

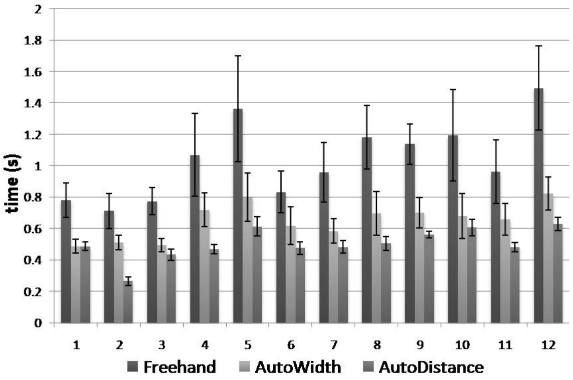

Figure 5: *corrective phase time: Freehand, AutoWidth vs AutoDistance*

5.4. Improved proportion in total movement time

For each user, total movement time of Freehand, AutoWidth and AutoDistance is averaged respectively. Figure 6 depicts the ratios by which the means of total movement time for AutoWidth and AutoDistance have been improved with respect to Freehand. The ratios are volatile among users, but AutoDistance always improves more than AutoWidth. The greatest improvement for AutoWidth and AutoDistance appears on user 5 whose aimed movements have on average progressed by 28.52% and 40.60% respectively. The least improvement comes from user 2 and user 1, still up to 12.42% and 19.26% correspondingly. After applying AutoDistance and AutoWidth in corrective phase, we are able to save subjects on average (without using ANOVA) 28.59% and 19.86% of total movement time, respectively.

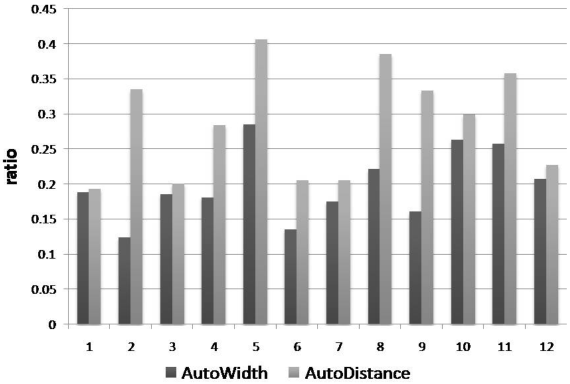

Figure 6: *AutoWidth vs AutoDistance: improved proportion for the means of total movement time wrt Freehand*

5.5. Improved proportion in ballistic phase time

A similar analysis has been done for improved proportion of AutoWidth and AutoDistance with respect to Freehand in the ballistic phase. Negative ratio indicates the proportion by which AutoWidth and AutoDistance regress. As can be seen from figure 7, user 1, 2, 3, 4, 6, 7 and 12 show almost no improvement and regression (within +/- 10%). The greatest improvement of AutoWidth comes from user 11 whose ballistic phase time has been improved by 20.95%, while that of AutoDistance comes from user 5, 30.63% improved. The greatest regression of AutoWidth and AutoDistance appears on user 9 and user 7, 5.32% and 6.97% regressed, respectively. The improved proportion is larger than regressed proportion, which, as mentioned in section *ballistic phase time*, may be due to the learning effect.

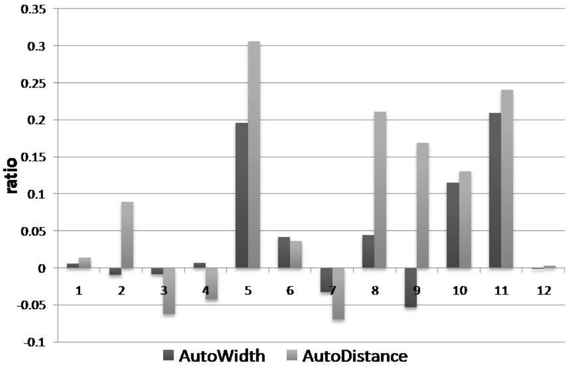

Figure 7: *AutoWidth vs AutoDistance: improved proportion for the means of ballistic phase time wrt Freehand*

5.6. Improved proportion in corrective phase time

A similar trend (figure 8) to improved proportion in total movement time can be found for the that in corrective phase time, except that the improved proportion is much greater. The greatest improvement of AutoDistance and AutoWidth has decreased corrective phase by 62.79% (user 2) and 45.03% (user 12), the least by 37.64% (user 1) and 25.91% (user 6) respectively. Generally speaking, AutoDistance still outperforms AutoWidth and they save on average (without using ANOVA) 50.21% and 36.68% of corrective phase time, respectively.

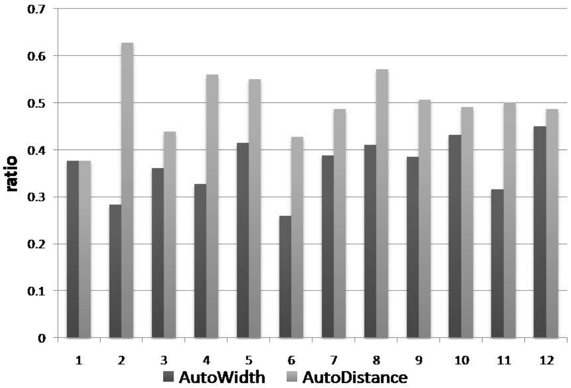

Figure 8: *AutoWidth vs AutoDistance: improved proportion for the means of corrective phase time wrt Freehand*

6. Conclusions

The proposed idea of combining two-component model and Fitts' law to reduce the index of difficulty of a 3D pointing task during the corrective phase provides effective strategies to improve the efficiency of 3D aimed movements. From the experimental results, we have shown that AutoDistance and AutoWidth are able to improve the efficiency of a selection task. The improvement does not play a part in reducing the time of the ballistic phase, but it significantly reduces the movement time of the corrective phase.

The AutoDistance and AutoWidth are parameter-dependent techniques which require different threshold settings for individuals. Also, both techniques rely on a prediction algorithm. For future work, we plan to develop parameter-independent selection techniques and more robust prediction algorithms.

7. Acknowledgements

We thank Chris Kruszynski for his advice concerning vtk and experiment setup and all subjects voluntarily participated in the experiment.

References

[AA08] ARGELAGUET F., ANDUJAR C.: Improving 3D selection in VEs through expanding targets and forced disocclusion. In *Proceedings of the 9th international symposium on Smart Graphics* (2008), pp. 45 – 57. 2

[Bal04] BALAKRISHNAN R.: "Beating" Fitts' law: virtual enhancements for pointing facilitation. *International Journal of Human-Computer studies 61* (2004), 857–874. 2

[BCR*03] BAUDISCH P., CUTRELL E., ROBBINS D., CZERWINSKI M., T P., BEDERSON B., ZIERLINGER A.: Drag-and-Pop and Drag-and-Pick: Techniques for accessing remote screen content on touch and pen-operated systems. In *Interact '03* (2003), pp. 57–64. 1

[BGBL04] BLANCH R., GUIARD Y., BEAUDOUIN-LAFON M.: Semantic pointing: Improving target acquisition with control-display ratio adaptation. In *CHI '04: ACM Conference on Human Factors in Computing Systems* (2004), pp. 519–526. 1

[Bie88] BIER E. A.: *Snap-Dragging: Interactive Geometric Design in Two and Three Dimensions*. PhD thesis, EECS Department, University of California, Berkeley, May 1988. 3

[Fit54] FITTS P. M.: The information capacity of the human motor system in controlling the amplitude of movement. *Journal of Experimental Psychology 47*, 6 (June 1954), 381–391. 1

[FKK07] FREES S., KESSLER G. D., KAY E.: PRISM interaction for enhancing control in immersive virtual environments. *ACM Transactions on Computer-Human Interaction (TOCHI) 14*, 1 (May 2007), Articale 2. 2

[GB04] GROSSMAN T., BALAKRISHNAN R.: Pointing at trivariate targets in 3D environments. In *Proceedings of the SIGCHI conference on Human factors in computing systems* (2004), pp. 447 – 454. 1

[IP96] IVAN POUPYREV MARK BILLINGHURST S. W. T. I.: The Go-Go interaction technique: non-linear mapping for direct manipulation in VR. In *Proceedings of the 9th annual ACM symposium on User interface software and technology* (1996), pp. 79 – 80. 2

[LvLNM09] LIU L., VAN LIERE R., NIEUWENHUIZEN C., MARTENS J.-B.: Comparing aimed movements in the real world and in virtual reality. In *VR '09: Proc. IEEE Virtual Reality 2009* (2009), pp. 219–222. 1, 2, 3

[MAK*88] MEYER D. E., ABRAMS R. A., KORNBLUM S., WRIGHT C. E., SMITH J. E. K.: Optimality in human motor performance: ideal control of rapid aimed movements. *Psychological Review 95* (1988), 340–370. 2, 3

[MI01] MURATA A., IWASE H.: Extending Fitts' law to a three-dimensional pointing task. *Human Movement Science 20*, 6 (Dec. 2001), 791–805. 1

[Smi96] SMITH W. J.: *Iso and ANSI Ergonomic Standards for Computer Products: A Guide to Implementation and Compliance*. Prentice Hall, Jan. 1996. 4

[Ste08] STEED A.: A simple method for estimating the latency of interactive, real-time graphics simulations. In *Proc. ACM VRST* (2008). 4

[SZZ06] SHAUGHNESSY J. J., ZECHMEISTER E. B., ZECHMEISTER J. S.: *Research Methods in Psychology*. McGraw-Hill, 2006. 5

[Woo99] WOODWORTH R. S.: *The Accuracy of Voluntary Movement*. Psychological Review, 1899. 1

[WS91] WARE C., SLIPP L.: Using velocity control to navigate 3D graphical environments: A comparison of three interfaces. In *Proceedings of the Human Factors Society 35th Annual Meeting* (1991), pp. 300–304. 1

[WWBH97] WORDEN A., WALKER N., BHARAT K., HUDSON S.: Making computers easier for older adults to use: Area cursors and sticky icons. In *CHI '97: Proc. SIGCHI conference on Human factors in computing systems* (1997), pp. 266–271. 1, 2, 3