Journal of Universal Computer Science, vol. 20, no. 7 (2014), 1026-1045
submitted: 31/1/14, accepted: 30/6/14, appeared: 1/7/14

First-Person Locomotion in 3D Virtual Environments: a Usability Analysis

Sergio Moya
(Polytechnical University of Catalonia - Barcelona - Spain
smoya@lsi.upc.edu)

Sergi Grau
(Polytechnical University of Catalonia - Barcelona - Spain
sgrau@lsi.upc.edu)

Dani Tost
(Polytechnical University of Catalonia - Barcelona - Spain
dani@lsi.upc.edu)

Abstract: 3D Virtual Environments (VE) are becoming popular as a tool for cognitive, functional and psychological assessment. Navigation in these environments is recognized as one of the most difficult activities in 3D Virtual Environments (VE). Users unfamiliar to 3D games, specially elder persons, get puzzled when they try to virtually move an avatar through these environments. Their inability to navigate prevents them from concentrating in the task and even to finish it. In this paper, we analyze the influence of different factors in locomotion control. We investigate the impact of having the cursor fixed at the camera center or leaving it free inside the current view. We also analyze the influence of the pitch angle on the camera control. In addition, we have designed an automatic locomotion system that we compare to user-controlled locomotion. We describe a virtual scenario and a test task that we have implemented to evaluate these different methods with users of diverse profiles.

We conclude that, for non-casual gamers, automatic locomotion is the faster and preferred mode, whereas gamers prefer to control locomotion themselves. For non-automatic locomotion, the performance is better when the cursor is fixed in the view center point during locomotion. On the contrary, for still positions, a free-cursor is preferable. Finally, restriction of the pitch angle in a solid angle that encompasses the objects related to the task is better than a free pitch angle rotation for still positions. During locomotion, fixing the pitch angle at the horizontal level enhances the locomotion.
Key Words: Virtual Worlds, Navigation, Locomotion, Camera Control, Usability
Category: H.2, D.2.8

1 Introduction

Serious Games (SG) are becoming a popular tool for learning [Freitas 2006], training professional activities [Graafland et al. 2012], as well as for motor and cognitive rehabilitation [Holden 2005, Cherniack 2011], and treatment of psychological disorders.

Three-dimensional virtual environments (VE) occupy a central role in leisure games and immersive virtual worlds. They provide a feeling of realism, of natural immersion that enforces players motivation. However, for "serious" applications, the use of 3D games is still limited, not only for technological and economical reasons, but because 3D environments are considered difficult to manage for a wide range of users [Bowman and Houges 1999, Ewais and de Troyer 2013].

In 3D VEs, the mapping between input actions and their effects in the 3D environment are not intuitive for non-expert users. In particular, the interaction control for navigation requires motor and sensitive skills that not all potential users have [Moffat 2009]. Although this control can sometimes be acquired through training, it often causes users' rejection from a start. Moreover, navigation interfaces require spatial cognitive resources that compete with those needed to perform the game tasks [Marsh et al. 2012].

Navigation consists of two components: way-finding and locomotion. Way-finding refers to the process of finding the way to the target location. It requires people to recognize the environment, determine where they are, how to get to a specific location [Darken and Sibert 1993]. Specific assistance mechanisms can be provided to ease way-finding avoiding users to get lost, for instance maps, world miniatures, arrows or indication panels [Burigat and Chittaro 2007]. Locomotion is the process of moving from the initial position to the desired one. In 3D games with a first-person perspective, locomotion consists of moving interactively the camera that represents the user's eyes, and in third-person approaches, it consists of moving the user's avatar. In the latter case, the camera tracks the user's avatar, and it is necessary to avoid it to collide with the avatar and other parts of the environment [Liu et al. 2011]. Mastering locomotion in a 3D leisure game requires a specific training, which is part of the challenge of playing, but that can be discouraging if training or rehabilitation are the purpose of the game.

Many SG consist of procedural tasks involving simple objects manipulation. For instance, cognitive rehabilitation tasks often reproduce daily life activities in virtual scenarios such as houses or shops. Professional training games recreate work-places where users learn to follow work procedures. In these games, users need locomotion to reach objects, but locomotion is not a goal of the game by itself. In this case, locomotion interferes with the foreseen development of the activity. It can hinder it, delay it, and even make it impossible for a large percentage of elderly or disabled persons [Grammenos et al. 2009]. Thus, simplified locomotion is needed in SGs with users with little computer literacy and even in leisure games when mastering locomotion is not part of the game challenge.

The objective of this work is to analyze how locomotion can be improved to make 3D VEs accessible to a wider range of users. We focus on first-person locomotion, because for most users it is easier to master than the third-person perspective. We have implemented a standard locomotion mode based on keyboard and mouse input, and different enhancements of this basic mode, including an automatic locomotion model that relieves users of the task of locomotion while keeping the benefits of manipulation in 3D. In order to evaluate users performance with these different modes, we have designed and implemented a usability test. We present an analysis of the results of these experiments.

2 Background

2.1 Free first-person locomotion

First-person locomotion (FPS) in 3D VEs requires to control various degrees of freedom (DOF) to reach the desired location and orientation. Using a pinhole camera model with the projection plane perpendicular to the view vector and with the projection window centered at the view reference point, these DOFs can be reduced to six assuming that the viewing distance and the camera opening are kept constant through locomotion. In FPS applications, these DOFs are expressed in terms of the camera position and three camera orientation angles.

The main difficulty of locomotion is to understand and apply efficiently the mapping between input devices and DOFs [Jankowski and Hachet 2013]. Many 3D leisure games such as World of Warcraft or Assassins Creeds use the mouse and the keyboard as the primary input devices. Using these two devices together it is relatively easy to control four degrees of freedom: two for the movement of the avatar in the floor controlled by four keys (arrows or ASWD), and two for the orientation of the head with the mouse. Thus, two DOFs must be removed: the head's roll, considered not very useful, and the vertical movement, because it is assumed that the user's avatar walks upon the floor [Parush and Berman 2004].

Other input devices such as joysticks are also used, but they are generally difficult to understand and manage by non-expert users [Bérard et al. 2009]. We herein focus on four-DOFS locomotion control based on mouse and keyboard interaction. The implementation of locomotion control with these devices is rather straightforward, thus, it has not been the object of technical discussions. However, according to our experience, the specific way of mapping user input onto the camera, has a strong impact on the user experience.

2.2 Assisted first-person locomotion

Reviewing existing computer games and related literature, we have found three main strategies to assist or avoid locomotion: to use a set of pre-computed 3D views, to use an extensible virtual arm and to make locomotion as automatic as possible. The former strategy uses a set of 2D images of the environment from different perspectives. All the objects with which the user may interact are visible at least in one view. Users change interactively the view to find one in which they are able to select the object of interest. This strategy is therefore 2D, but it gives the illusion of 3D. This approach is often used in *escape games*. It has the advantage of removing the technological barrier of locomotion, but it gives a low sensation of immersion.

The second strategy, called the *Go-Go* technique, [Poupyrev et al. 1996] dynamically expands the user's arm length in order to reach objects. The arm's length function is linear when the user wants to reach objects that are nearer than the real arm length and non-linear when the user wants to select non reachable objects. Thus, it is not necessary to move the camera to bring it closer to the object. Users only need to modify its orientation. In contrast, a new DOF is added: the arm's length. The control of this DOF is not very intuitive, because it has no correspondence with a real world action. In addition, far objects may have a small projected area, and thus they can be difficult to select. Moreover, the density of the environment must be low, otherwise occlusions between objects may prevent the selection of some of them.

Finally, the third approach consists of navigating automatically to a target location or *Point of Interest (POI)*. It involves three problems: to determine the target location and orientation, to compute the path from the current position to the target and to find a balanced and natural way of following this path. The target location can be defined interactively [Hachet et al. 2008] or determined automatically. In the latter case, the techniques designed to compute automatically the best camera placement for the exploration of virtual environments [Argelaguet and Andujar 2010] and scientific data [Bordoloi and Shen 2005] can be applied. However, automatic determination of the POI avoids not only locomotion but also target-finding and way-finding. This reduces a lot the set of training tasks that can be designed. Thus, automatic camera placement is mostly provided at the beginning of the game or at the transition between scenarios [Christie and Olivier 2008] to preserve the continuity of the game-play, while providing the best view of the environment.

Once the target point has been determined, two main approaches are applied to compute the path: reactive and indirect methods [Jankowski and Hachet 2013]. Reactive methods use autonomous robotics strategies to drive the camera through the shortest possible path, avoiding obstacles [Reese and Stout 1999]. They apply to the camera the navigation models that are used for the animation of autonomous non-player characters [Hartley and Mehdi 2012]. Indirect approaches translate users needs into constraints on the camera parameters, which they intend to solve [Driel and Bidarra 2009].

3 Locomotion methods

Through a five-years period of time in the frame of a SG development project for neurorehabilitation [Tost et al. 2009], we observed the user's behavior while performing the task of reaching objects within a 3D VE. The locomotion control of these SGs was based on the 4-DOFs mouse and keyboard model described in Section 3.1. We differentiated three stages in the task from the start point to the selection of the target object: the recognition stage, in which users perform the initial target search, the motion stage, which includes path finding and navigation to the target, and the final reorientation stage to precisely point onto the object. In general, during the recognition stage and in the final camera reorientation, users keep the camera position still and only modify its orientation. During the movement stage, they modify mostly the camera's position, with some occasional pauses, and a little less its orientation. We detected that users had difficulties in orientating the camera to focus on the target object and in controlling the pitch angle, in the three stages in general and specially during the motion stage. In order to address these problems, we propose two alternative ways of controlling both parameters (see Sections 3.2 and 3.3). Thus, we analyze four locomotion modes that combine the two ways of controlling these parameters: *centered-cursor*, *centered-cursor with restricted-pitch*, *free-cursor* and *free-cursor with restricted-pitch*. In addition, we introduce a fully *automatic-locomotion* to compare with the four user-driven methods.

3.1 Basic locomotion

The *basic mode* of locomotion is the classical first person four degrees of freedom walking metaphor. The camera control is done by mapping the mouse movement to specify the yaw and pitch angle to control the camera orientation, and the keyboard arrows in order to control forward, backward, left and right movements. The pitch angle is restricted within a parameterized range to allow looking at the floor and ceiling, but it forbids complete turns. The movement has constant speed except for a short acceleration at its beginning and deceleration at its end.

3.2 Camera orientation

We analyze two different ways of controlling the camera orientation using the mouse: having the cursor always fixed in the center of the view (*centered-cursor*), and leaving the cursor free inside the current view (*free-cursor*). In the first approach, when a user moves the mouse, the view changes applying yaw and pitch rotations and keeping the cursor in the center of the view. Therefore, users are only able to select the objects that are in the center of the view. Involuntary movements of the cursor yield to a camera reorientation. In the second approach, users can move the cursor freely without moving the camera. Thus, they can select any object in the view, be it centered or not. To change camera orientation, users move the cursor towards the border of the view.

3.3 Pitch restriction

In the *basic mode*, where the pitch angle is only restricted in the range [-50, 50], non-expert users have difficulties in controlling pitch rotation, even at still positions. Pitch-control is even more complex during locomotion. Non-expert users find it more difficult due to the combination with camera motion. We have observed that they tend to make exaggerate movements, and they often focus at the floor or at the ceiling while walking, because they have difficulties in recovering the horizontal position. Because of that, they often get disoriented. We have proposed the *restricted-pitch* mode that limits the pitch camera orientation depending on the positions of interactuable objects. Specifically, we compute the solid angle defined by the camera position and the objects with which the user can interact. We restrict pitch to this solid angle, clipped in the range [-50, 50], but extended to the horizontal direction. Thus, when interactuable objects are on the floor, the pitch angle will be at most [-50, 0].

3.4 Automatic locomotion

In the *automatic locomotion*, users indicate where they want to go, and they are automatically driven there. This way, the focus is put on the destination and not on the path towards it, and locomotion is decoupled from the training objectives of the SG. This strategy can be implemented in two ways: by computing only the final camera position, or by calculating all the camera path towards this position. In the first case, the transition from one view to the next is very abrupt. Therefore, we reserve it for the transition between one scenario to the other. In this work, we focus on the second mode. We compute all the camera path and orientation to provide users a feeling of realism and immersion,

To indicate the target's location, users click onto it. If the target location is reachable from the avatar's position, i.e. if it is at a smaller distance than the avatar's arm estimated length, the system interprets the user click as a selection. However, if the object is not reachable, the system interprets that the user wants to go towards it, and the system computes the corresponding path and follows it automatically. If the target object is not visible from the current camera's position, users indicate movement by steps, giving a first path direction, stopping the movement to reorient the camera and clicking again to specify a new direction. For instance, to reach an object in another room, users will need first to reach corresponding door. Thus, although locomotion is removed, way-finding tasks are still feasible.

4 Implementation

The software is implemented on top on Blender Game Engine (BGE). The implementation of the basic locomotion mode and the assistance mechanisms is straightforward using BGE API.

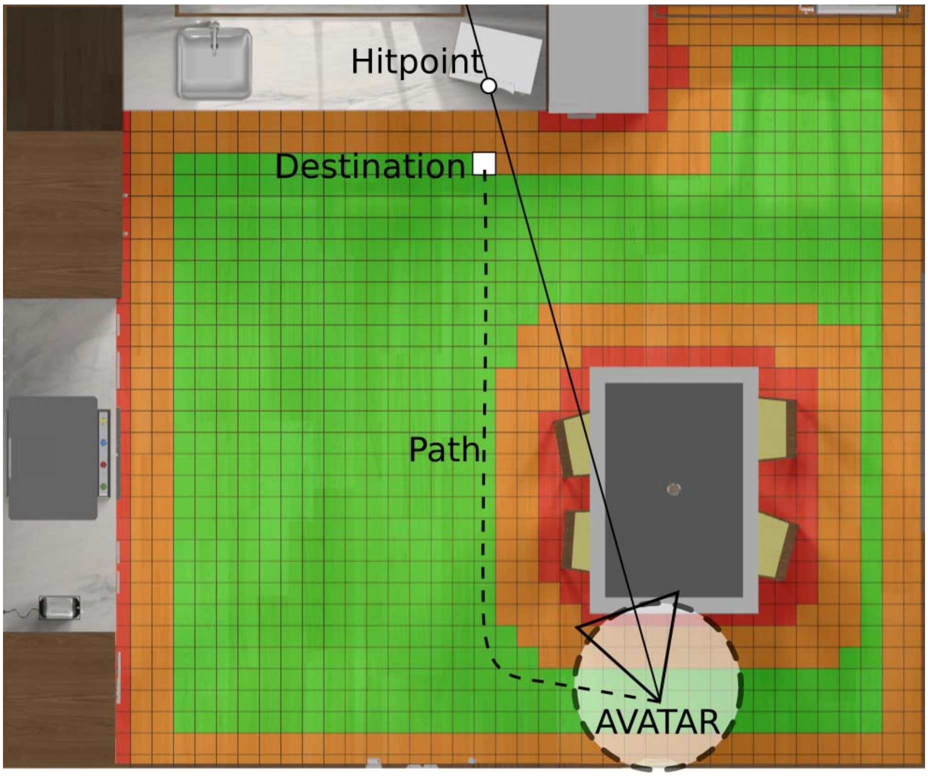

Figure 1: Example of the locomotion strategy. The system use the floor's grid to determine the best path to reach the hit-point. The destination cell is draw in white, and the cells are classified as occupied (red), unreachable (orange) and reachable (green).

The system uses the target object's position to determine the best avatar's location to reach it. This location lays on a navigation grid on top of the VE's floor. Figure 1 shows a grid example of a kitchen. The cells of the floor can be classified as occupied (red), unreachable (orange) or reachable (green). The unreachable cells are free cells where the avatar cannot go, because it would collide with other elements of the VE. Thus, the avatar is only allowed to be in a position inside reachable cells. The system uses the grid to determine which of the reachable cells is the best to interact with the target object. The naive strategy consists of finding the closest cell to the target. However, it does not take into account the possible occlusions. Therefore, we choose the closest cell from which the target is visible. We compute these cells in a pre-process, casting rays from the surfaces cells to virtual camera positions centered at the grid floor cells. Our scenario model stores these cells associated to each surface cell. Then when the system wants to determine the best destination for a target, it selects the closest cell that belongs to the set of the target's surface cell. Taking into account that the objects during the task can change their positions, it is possible that all the cells are occupied, and then the avatar cannot reach the target. In those cases, the system's logic is the responsible of asking the user to move some objects to be able to reach the target.

Once the system has the current position of the avatar and the destination position, it computes the path that allows the avatar to move inside the VE without colliding with any object. The method used is an implementation of the A* path-finding method that minimizes the Euclidean distance, and uses the floor's grid to compute a discrete path. After this process, the system computes a Bézier path that follows the discrete path computed before. This new path allows the system to perform softer movements and keep a constant speed.

5 Analysis methodology

5.1 Tests users

To test the locomotion approaches we have designed a task consisting of pointing onto three objects strategically placed in a 3D environment. We have studied the locomotion approaches with 44 users (26 men, 18 women) stratified acording to their genre and age (from 16 to 81 years-old), being the maximum error for a confidence level of 95% of 14,8%. Figure 2 shows the distribution according to age and genre. The test users were volunteers, not paid for the experiment. They were recruited through the authors social network, and they did not have previous experience with the software. Test users were classified into two categories: casual gamers (18) and non-casual gamers (26). All the more than 50 years-old users were non gamer, but all except 2 had some skills using computers.

5.2 Data collection

In order to make a systematic analysis of the results, the software implemented for the tests stored in a file all the interactions done by the user in terms of mouse and keyboard movement events. In addition, it stored camera positions and orientations with a frequency of 10 times per second. The software also recorded all the frames in which a mouse-click user event happened. This way, we were able to rebuild the sessions from the recorded frames and data. The analyses described below are based on a processing of these data.

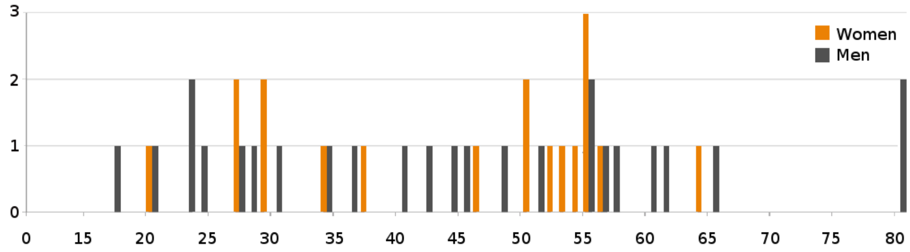

Figure 2: Histogram of users distribution according to age and genre. In the x-axis, the ages. In the y-axis the number of users of this age.

5.3 Testing procedure

Each test was performed individually, in presence of a facilitator, using a laptop with an external mouse. The test began with the facilitator explaining the aims of the experiment and filling a questionnaire in which users were asked about their age, skills in using computers and computer games preferences and frequency of use.

The facilitator first made a demonstration of the environment, by navigating through it, describing it and asking questions to the user about its composition (What do you think this object is? Could you tell me where is the washing machine? and so on), to verify that the virtual environment was well understood by users. During this tour, the facilitator explained the concept of cursor and locomotion. The tour was skipped for casual gamers who already knew the concept of 3D environment and locomotion. After that, non-casual gamers were allowed to try to navigate for a short lapse of time with the five modes to understand the relationship between the input device and virtual environment. Again, some casual-gamers skipped this stage. Next, all users performed the tests, and at the end, in the post-test stage, they were asked to fill the SUS usability questionnaire, and express freely their opinions. The facilitator took notes on the behavior and comments of the users throughout the experiment.

5.4 Tasks description

The chosen scenario is a loft (see Figure 3). We set two tasks. In the first task, the user's avatar initial position was in front of the window. Users were asked to select three small objects (a bottle of water, an oilcan, a cake box) located in the kitchen part of the loft, in front the window. The difficulty of the task was that there were two obstacles to avoid (the tables), one of them, the low table,

Figure 3: Two views of the scene. The targets are at different heights relative to the ground.

Figure 4: Test tasks (top view of the scenario). Task 1 (left): it consists of reaching the bottle of water dashed in blue (1), the oilcan in yellow (2) and, finally, the cake box in red (3). Task 2 (right): it consists of reaching the refrigerator dashed in blue (1), the bed in yellow (2) and the window in red (3). The ideal paths to reach each of the targets are shown with dashed lines.

out of the initial view fustrum (Figure 4 left). The three objects were located at different heights relative to the ground in order to force the user to look at different target heights while they are walking. In the second task the user's avatar initial position was at the entrance door. The three targets were large objects (window, refrigerator and bed). A click in any point on these objects

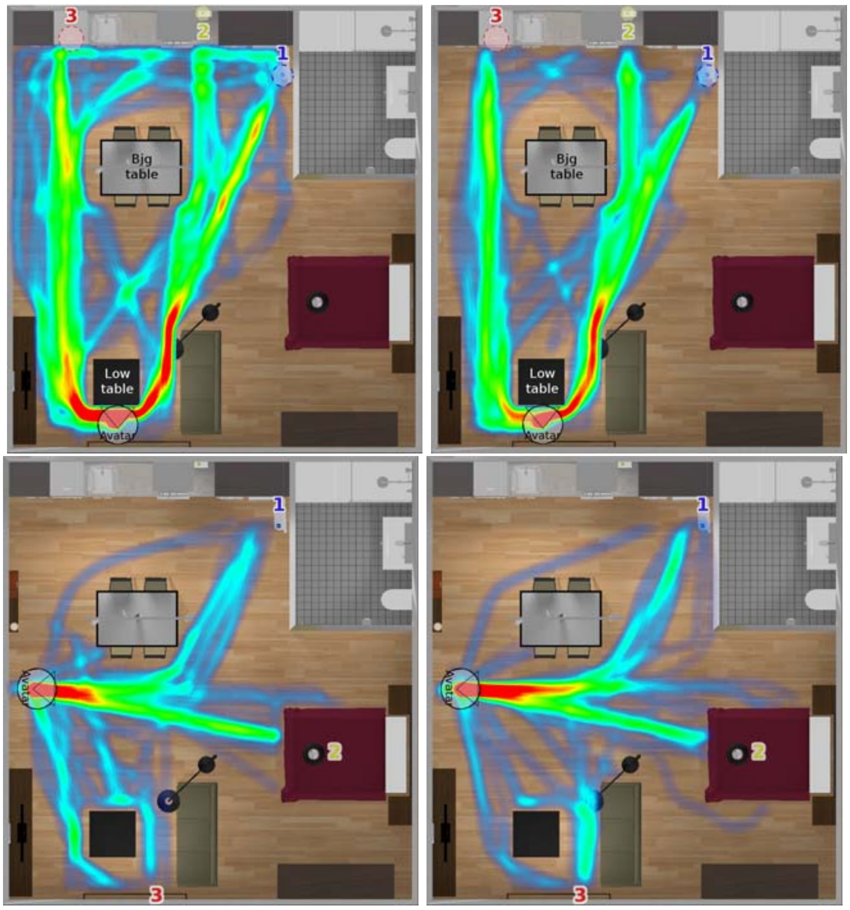

Figure 5: Paths in task 1 (top) and task 2 (bottom) with regular pitch (left) and restricted-pitch (right) modes.

from a reachable distance was enough to select them. The paths were shorter and with less obstacles than in the first task (see Figure 4 right).

The two tasks were segmented in three stages, one per object. The entire tasks were completed 5 times by each user, one per method. In Figure 4, the initial positions of the avatar in the environment before performing any interaction is represented by a solid line circle with a triangle inside that indicates the camera. The three dashed line circles represent the considered as ideal final positions. The dashed lines represent the ideal paths, i.e those that require the minimum distance to reach objects and the minimum variation of vertical angle of head movement.

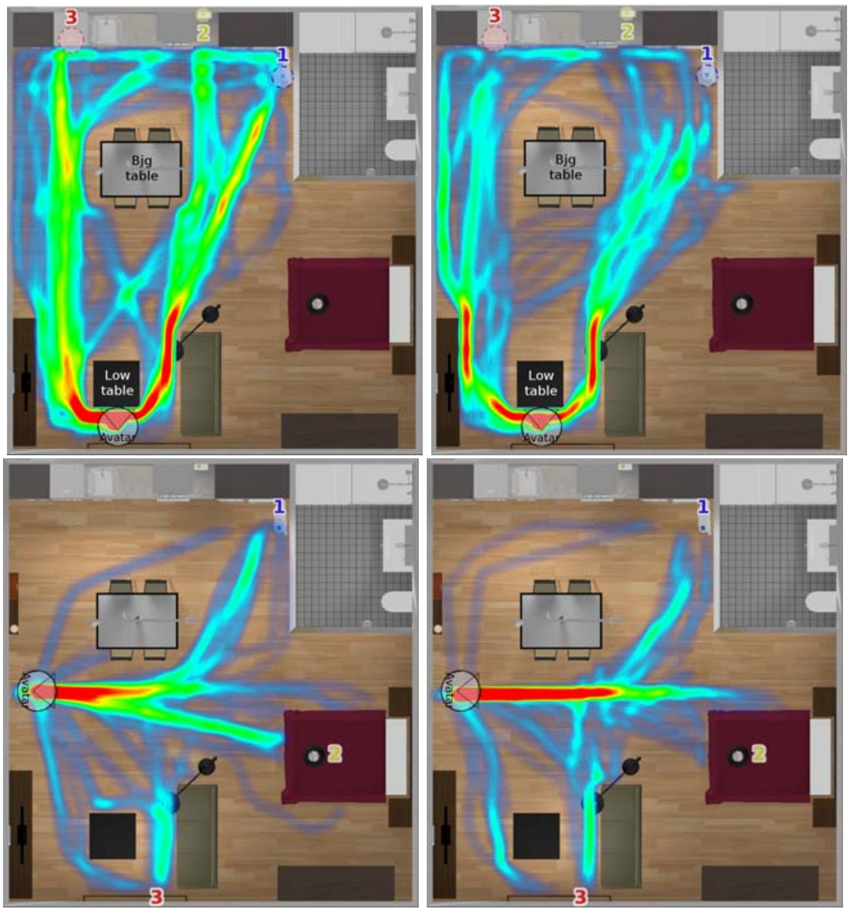

Figure 6: Paths in task 1 (top) and task 2 (bottom) with *centered-cursor* (left) and *free-cursor* (right).

6 Results and discussion

6.1 Qualitative analysis of the paths

Using the data stored during the tests, we reconstructed the paths followed by users in the two tasks and the four non-automatic modes. We show them in Figures 5 and 6, using a temperature range color encoding. The more transited is the place, the higher temperature it represents. In other words, the places where more users pass are colored in red.

Figure 5 represents the paths with the regular pitch mode (left) and the *restricted-pitch* mode (right), and Figure 6 represents the paths with the *centered-cursor* (left) and *free-cursor* (right). Figure 5 shows that the *restricted-pitch* yields to shorter path. In particular, in task 1, it avoids unneeded movements between the location of target 2 and 3, clearly perceptible in the tests with regular pitch. With the *restricted-pitch*, users tended to go rather straightforwardly to their target destination.

From Figure 6, it can be seen that *free-cursor* during locomotion yields to more path distortion and unnatural paths. Specifically, for the first stage of the first task, users tended to disperse towards the bedroom, and, for the third stage, they moved stuck to the wall, a quite weird motion.

The differences between modes are less significant in the second task than in the first one. This is consistent with the numerical results discussed next. This is due to the fact that the task is simpler and the paths shorter.

6.2 Qualitative analysis

To evaluate the different modes and compare them with the automatic locomotion, users ranked the locomotion and camera orientation methods by answering a questionnaire that ranked from 0 to 5 the level of difficulty and pleasantness (see Table 1). All users, gamers and non-gamers, found the automatic locomotion very easy. It is also the preferred mode for non-gamers and elder users. Most elder users made comments such as: "This is really better", "This way yes, I can do it", "Now, I'll be able to really do the task fast". Young gamers preferred to have the control of the camera but the automatic mode did not displeased them. Comments of this group of users were: "It's nice, but it's only useful for persons that don't play. I wouldn't use it. No real game would. I prefer to have the control."

All users coincide on finding the *free-cursor* with locomotion the most difficult method. They did not like it. '"It's tiring. You loose the concentration on the environment. I'm not sure about where and when I start to rotate" were some of their comments. The *free-cursor* on still positions was preferred by non-gamers ("It is easier to control, I master better the navigation") , but not by gamers ("It's unpractical. This is not the usual way"). The restricted-pitch was difficult to appreciate by gamers, who scored this mode as average (2.5) meaning that it was indifferent for them (" I didn't notice the difference"). By opposite, non-gamers showed a preference for restricted-pitch ("I think that I did it better this way"). In general, for non-gamers the level of pleasantness is inversely proportional to the level of difficulty.

6.3 Quantitative analysis

In order to analyse the results quantitatively, for all users and the three paths, we have computed the distance travelled by the camera, the pitch angle at still poitions and the time spent in locomotion and at still positions.

Qualitative results for non-gamers		
	Difficulty	**Pleasantness**
Stopped w/ centered-cursor	2.7	3.0
Stopped w/ free-cursor	3.0	3.5
Stopped w/ regular pitch	3.0	3.0
Stopped w/ goal-restricted pitch	2.5	4.5
Walking w/ centered-cursor	2.0	4.0
Walking w/ free-cursor	4.5	1.0
Walking w/ regular pitch	3.4	3.4
Walking w/ fixed pitch	3.2	4.2
Automatic locomotion	0.0	4.8

Qualitative results for Gamers		
	Difficulty	**Pleasantness**
Stopped w/ centered-cursor	0.0	4.0
Stopped w/ free-cursor	0.5	3.5
Stopped w/ regular pitch	0.0	2.5
Stopped w/ restricted-pitch	0.0	2.5
Walking w/ centered-cursor	0.0	4.0
Walking w/ free-cursor	1.3	2.6
Walking w/ regular pitch	0.0	2.5
Automatic locomotion	0.0	3.0

Table 1: Qualitative results for non-gamers (top) and gamers (bottom). Average score from 0 to 5.

We have removed from the analysis 3 samples that falled out of the confidence interval. Thus the actual number of analyzed results is based on a sample number of 41 users.

Figure 7 shows the value of distance and time obtained by the different users with *free cursor* (left) and *centerer cursor* (right), and depicts the information representing users as dots in a cross diagram distance versus time. Figure 8 shows the value of distance and time spent at still position with and without restricted pitch.

Table 2 shows the results with *centered-cursor* and *free-cursor* and the values of the ideal paths to perform the tasks. Table 3 shows the average and the standard deviation of these two parameters. It also shows the average and standard deviation of the time spent at still positions and the pitch angle. We have computed the average gain in distance, time and angle of one technique in

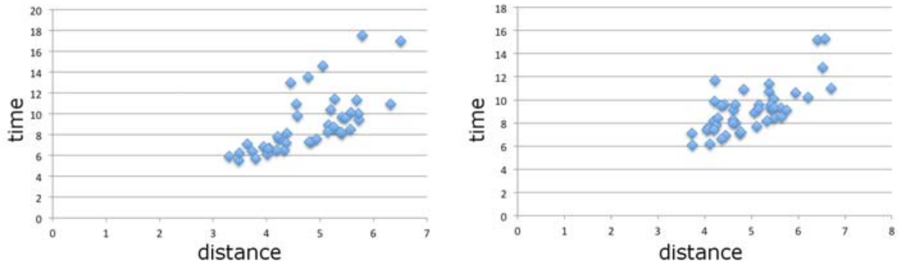

Figure 7: User results with *free-cursor* (left) and *centerer-cursor* (right). In the x-axis, the travelled distance expressed in normalized units of the VE which are equivalent to meters. In the y-axis, the time needed for travelling expressed in seconds. Each dot represents a user.

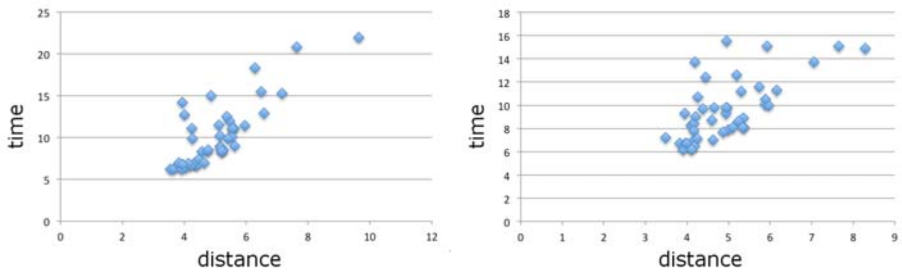

Figure 8: User results: at left, with restricted pitch and at right with regular pitch. In the x-axis, the travelled distance expressed in normalized units of the VE which are equivalent to meters. In the y-axis, the time needed for travelling expressed in seconds. Each dot represents a user.

comparison to the other.

It can be seen that in both tasks and specially in the more complex task (task 1) restricted-pitch speeds up performance at still positions as well as during walking and that the gain is higher at still positions. When the avatar is stopped, the pitch restriction avoids rotations that hide the target from the current view. Thus, users focus faster on the target. During walking, the pitch restriction avoids unnatural rotations that cause disorientation.

The second column shows the distance traveled, the third the time spent, the fourth the time spent in still position, and the fifth and sixth the pitch and yaw rotation accumulated for each method. It can be seen that when the avatar stays still, the *free-cursor* obtains better results that the *centered-cursor*: it requires less distance, time and rotation angles. Again the gain is more significant in

Cursor - task 1					
	Walking		Stopped		
	Dist. (m)	Time (s)	Time (s)	Pitch (deg.)	Yaw (deg.)
Ideal	6.56	5.33	-	-	-
Centered	7.07 | 1.46	9.4 | 3.96	17.46 | 23.27	102.6 | 85.19	40.7 | 27.55
Free	7.63 | 2.95	12.28 | 8.64	14.05 | 20.01	37.2 | 23.79	5.2 | 3.27

Cursor - task 2					
	Walking		Stopped		
	Dist. (m)	Time (s)	Time (s)	Pitch (deg.)	Yaw (deg.)
Ideal	4.15	6.37	-	-	-
Centered	4.93 | 0.73	8.88 | 1.52	8.22 | 7.81	44.20 | 81.60	15.19 | 23.86
Free	5.09 | 1.60	9.79 | 4.87	10.12 | 12.39	11.69 | 24.29	1.15 | 5.43

Table 2: Users results for task 1 (top) and task 2 (bottom) with *centered-cursor* and *free-cursor*. The ideal row shows the minimum distance and time needed to reach the targets. The rest of cells show the average value and the standard deviation value. Distances in meters at the VE scale, time in real seconds and angles in degrees are separated between still positions and movement.

Pitch - task 1				
	Walking		Stopped	
	Dist. (m)	Time (s)	Time (s)	Pitch (deg.)
Ideal	6.56	5.33	-	-
Regular	7.53 | 2.13	12.51 | 7.93	29.73 | 19.59	163.29 | 89.5
Restricted	7.07 | 1.46	9.4 | 3.96	17.46 | 9.27	102.56 | 85.19

Pitch - task 2				
	Walking		Stopped	
	Dist. (m)	Time (s)	Time (s)	Pitch (deg.)
Ideal	4.15	6.37	-	-
Regular	5.15 | 1.28	11.22 | 5.67	13.14 | 13.73	18.19 | 27.09
Restricted	4.93 | 0.73	8.88 | 1.52	7.34 | 6.94	11.69 | 24.29

Table 3: Users results or task 1 (top) and task 2 (bottom) with regular pitch and restricted-pitch. The table shows at each cell the average value and the standard deviation value, except for the Ideal row that shows only the ideal value. Distances in meters at the VE scale, time in real seconds and angles in degrees are separated between still positions and movement.

Task 1 than in Task 2. The gain can be explained by the fact that users need to perform less rotations to select the targets, because if the target is visible in the view fustrum, users can click on them directly by moving the cursor onto the window without moving the camera. With the *centered-cursor*, instead, the target needs to be in the center of the view where the cursor is located to be selected. This yields to more rotations (64% more pitch rotation and 87% more yaw rotation) and more corrective movements. On the contrary, in Task 1, when the avatar moves, the *free-cursor* requires more time than the *centered-cursor*. This is consistent to the fact that the yaw rotation is slower in this mode, because users need to put the cursor on the borders of the screen to rotate. The distance are also shorter with *centered-cursor*, but the differences are very small. In Task 2, the times and distance are similar in both modes, the time is even slightly better with *free-cursor*. This can be explained by the fact that the targets were very large in comparison to those of Task 1 and could be selected by clicking at any point of their projected area, so users did not need to rotate the camera in yaw to perform selection.

In Table 4, we present the results of the test segmenting users into two categories according to three different classification criteria: gender, age and experience in 3D games. The values shown in the table are the average and standard deviation values in distance, time and angles of users with the corresponding profile. During walking, the differences between groups are similar in the *centered-cursor* mode and in the *free-cursor* mode, but at still positions, the differences are stronger. In general, the variation of distance between groups is not high. On the contrary, there are significant variations of time and angles. As expected, elder users were always slower than young ones, and they made larger pitch and yaw rotations. The same happens with non-gamers and gamers. In relation to gender, men were also slower than women, they traveled more distance and made larger rotations.

The highest difference in time between the two age groups is at still positions with *free-cursor*, because younger users required less yaw rotation (all of them were under the average yaw rotation) and were more agile and fast at moving the cursor to the graphical area boundary. On the contrary, the highest difference in time between the two gender groups is at still positions with *centered-cursor*. It seems that women had more difficulties in putting the target object at the center of the view, but that their agility was similar to that of men.

We also analyzed the influence of training in the performances. Table 5 shows the gain in time and distance between the realization of the task at the beginning and the end of the training stage. As expected, non-gamers had a higher gain.

In the automatic locomotion system, 80% of users for Task 1 and 90% of users for Task 2 hit the target object at first. The percentage is higher in Task 2, because the target objects were larger. Errors came from a lack of precision in the

Centered-cursor					
	Walking		**Stopped**		
	Dist. (m)	**Time (s)**	**Time (s)**	**Pitch (deg.)**	**Yaw (deg.)**
Men	5.25 | 0.75	9.70 | 2.21	9.14 | 13.86	22.23 | 31.09	7.36 | 8.87
Women	4.79 | 0.75	8.73 | 1.73	9.72 | 8.34	58.28 | 102.16	20.56 | 29.40
< 50	4.91 | 0.66	8.21 | 1.20	6.34 | 6.06	23.78 | 41.03	8.94 | 11.02
≥ 50	5.06 | 0.86	9.79 | 2.18	11.55 | 14.15	54.58 | 96.96	18.53 | 28.50
Gamer	4.92 | 0.66	8.51 | 1.14	5.84 | 4.99	31.65 | 51.41	9.41 | 12.56
Non-gamer	5.07 | 0.88	9.73 | 2.40	12.63 | 14.78	51.54 | 99.13	19.31 | 29.43

Free-cursor					
	Walking		**Stopped**		
	Dist. (m)	**Time (s)**	**Time (s)**	**Pitch (deg.)**	**Yaw (deg.)**
Men	4.83 | 0.74	8.94 | 2.75	9.84 | 12.43	10.58 | 23.20	0.00 | 0.00
Women	4.76 | 0.86	9.02 | 3.00	8.41 | 9.01	13.63 | 26.38	2.19 | 7.43
< 50	4.56 | 0.71	7.41 | 1.36	6.24 | 5.17	11.40 | 22.55	0.99 | 4.05
≥ 50	4.96 | 0.84	10.16 | 3.12	11.17 | 12.79	12.89 | 26.72	1.36 | 6.55
Gamer	4.72 | 0.68	7.99 | 1.52	6.37 | 4.85	15.99 | 28.32	0.85 | 3.75
Non-gamer	4.85 | 0.92	9.98 | 3.50	11.75 | 13.43	8.51 | 20.57	1.56 | 7.00

Table 4: Comparative results according to users profiles for *centered-cursor* (top) and *free-cursor* (bottom) at still positions (left) and during movement (right). The values represent the average and the standard deviation for each user profile.

Training						
	Average		**Gamer**		**Non-gamer**	
	Dist.	**Time**	**Dist.**	**Time**	**Dist.**	**Time**
Gain	2%	11%	1%	5%	3%	14%

Table 5: Influence of users training. Differences in distance and time between the first training step and the tasks.

selection when users hit on another object nearby the target. In this case, with the *free-cursor* mode, they could still select the target in most cases, because it was inside the view. Thus, they did not interact further with the application. Otherwise, they needed a short locomotion to the target.

7 Conclusions

The motivation to this paper is to analyze how to make locomotion easier for 3D SGs. Based on the observation of how non-casual gamers navigated, we have proposed methods to alleviate the problems of camera orientation during movements and at still positions, and a fully automatic locomotion mode. We have tested these methods with non-casual gamers as well as gamers. The results have shown that automatic locomotion can be good solution for both types of users in most games whose goal is not locomotion itself. When users become familiar with the game, they prefer to have the control of the camera. Then the *free-cursor* and *restricted-pitch* at still positions can enhance the efficacy of players. During the movement, the *centered-cursor* and the regular pitch are preferable.

Starting from this preliminary analysis, we plan to generalize the tests on basis of a free web application, in order to attract more users, and collect results through time. In addition, we are already investigating how to automatically detect when user are disoriented, and consequently help them with smooth camera re-orientation towards the target. Finally, we plan to analyze locomotion on a touch screen devices.

Acknowledgments

This work has been partially funded by projects: TIN2011-24220 and Rehabilita from the Ministerio de Economía y Competitividad.

References

[Argelaguet and Andujar 2010] Argelaguet, F., Andujar, C.: "Automatic speed graph generation for predefined camera paths"; Smart graphics, 2010, 115-126.

[Bérard et al. 2009] Bérard F., Ip J., Benovoy M. El-Shimyand D., Blum J., Cooperstock J.R: "Did Minority Report Get It Wrong? Superiority of the Mouse over 3D Input Devices in a 3D Placement Task"; Proc. Interact'09, Lect. Notes in Comp. Sci. 5727 (2009), 400-414.

[Bordoloi and Shen 2005] Bordoloi U., Shen H.-W.: "View selection for volume rendering"; Proc. IEEE Visualization'05; 52-62, 2005.

[Bowman and Houges 1999] Bowman D., Houges L.: "Formalizing the design, evaluation, , application of interaction techniques for immersive virtual environments"; Journal of Visual Languages and Computing, 10, 1999, 37-53.

[Burigat and Chittaro 2007] Burigat S., Chittaro L.: "Navigation in 3D virtual environments: Effects of user experience and location-pointing navigation aids"; Int. J. Hum.-Comput. Stud., 65, 2007 945-958.

[Cherniack 2011] Cherniack E.: "Not just fun and games: applications of virtual reality in the identification and rehabilitation of cognitive disorders of the elderly"; Disabil. Rehabil. Assist. Technol., 6, 4, 2011 283-289.

[Christie and Olivier 2008] Christie M., Olivier P.: "Camera control in computer graphics"; Computer Graphics Forum, 27, 8, 2008 1-197.

[Darken and Sibert 1993] Darken R., Sibert J.: "A toolset for navigation in virtual environments"; Proc. ACM symposium on User interface software and technology, 1993 157-165.

[Driel and Bidarra 2009] Driel L., Bidarra R.: "A semantic navigation model for video games"; Proc. of the 2nd International Workshop on Motion in Games 2009, 146-157.

[Ewais and de Troyer 2013] Ewais, A. and De Troyer O.:"Usability Evaluation of an Adaptive 3D Virtual Learning Environment"; Int.JOurnal of Virtual and Personal Learning Environmenst, 4 (1), 2013 16-31.

[Freitas 2006] Freitas S.: "Learning in immersive worlds"; Tech. Rep., JISC e-learning programme, 2006.

[Graafland et al. 2012] Graafland M., Schaagen J., Schijven M.: "Systematic review of serious games for medical education and surgical skills training"; British Journal of Surgery, 1, 99, 2012 1322-1330.

[Grammenos et al. 2009] Grammenos D., Savidis A., Stephanidis C.: "Designing universally accessible games"; ACM Computers in Entertainments, 7, 1, 2009 1-29.

[Hachet et al. 2008] Hachet M., Decle F., Knodel S., Guitton P.: "Navidget for Easy 3D Camera Positioning from 2D Inputs"; Proc. IEEE Symposium on 3D User Interfaces 2008, 83-89.

[Hartley and Mehdi 2012] Hartley T., Mehdi Q.: "In-game adaption of a navigation mesh cell path."; Proc. Int. conference on computer games, 2012 230-236.

[Holden 2005] Holden M.: "Virtual environments in motor rehabilitation: a review."; Cyberpsychology & Behavior, 8, 3, 2005 187-211.

[Jankowski and Hachet 2013] Jankowski J., Hachet M.: "A survey of interaction techniques for interactive 3d environmnents"; Eurographics 2013 STAR 65-93.

[Liu et al. 2011] Liu L., Jean-Bernard J., van Liere, R.: "Revisiting path steering for 3d manipulation tasks". Int. Journal Hum.-Comput. Stud., 69, 2011 170-181.

[Marsh et al. 2012] Marsh W.E., Putman, M, Kelly, J.W., Dark V.: "The cognitive implications of semi-natural virtual locomotion"; Virtual Reality Short Papers and Posters (VRW), 2012.

[Moffat 2009] Moffat S.: "Aging and spatial navigation: what do we know and where do we go?"; Neuropsychology review, 19, 4, 2009 78-89.

[Parush and Berman 2004] Parush A., Berman D.: "Navigation and orientation in 3d user interfaces: the impact of navigation aids and landmarks"; Int. Journal of Human-Computer Studies, 61, 3, 2004 375-395.

[Poupyrev et al. 1996] Poupyrev I., Billinghurst M., Ichikawa S.W.T.: "The Go-Go interaction technique: non-linear mapping for direct manipulation in VR"; Proc. 9th annual ACM symposium on User interface software and technology, 1996, 79-80.

[Reese and Stout 1999] Reese B., Stout B.: "Finding a pathfinder"; Symposium on arificial intelligence and computer games, 1999 69-72.

[Tost et al. 2009] Tost D., Grau S., Ferré M., Garcia P. et al.:"PREVIRNEC: A cognitive telerehabilitation system based on Virtual Environments"; Proc. IEEE Virtual Rehabilitation, 2009 87-93.