

We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,900

Open access books available

186,000

International authors and editors

200M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com



Personalization of Virtual Environments Navigation and Tasks for Neurorehabilitation

Dani Tost, Sergi Grau and Sergio Moya
*CREB-Polytechnical University of Catalonia
Spain*

1. Introduction

The use of “serious” computer games designed for other purposes than purely leisure is becoming a recurrent research topic in diverse areas from professional training and education to psychiatric and neuropsychological rehabilitation. In particular 3D computer games have been introduced in neuropsychological rehabilitation of cognitive functions to train daily activities. In general, these games are based on the first person paradigm. Patients control an avatar who moves around in a 3D virtual scenario. They manipulate virtual objects in order to perform daily activities such as cooking or tidying up a room. The underlying hypothesis of Cognitive Neuropsychological Virtual Rehabilitation (CNVR) systems is that 3D Interactive Virtual Environments (VE) can provide good simulations of the real world yielding to an effective transfer of virtual skills to real capacities (Rose et al., 2005). Other potential advantages of CNVR are that they are highly motivating, safe and controlled, and they can recreate a diversity of scenarios (Guo et al., 2004). Virtual tasks are easy to document automatically. Moreover, they are reproducible, which is useful for accurate analyses of the patients behavior.

Several studies has shown that leaving tasks unfinished can be counterproductive in a rehabilitation process (Prigatano, 1997). Therefore, CNVR systems tend to provide free-of-error rehabilitation tasks. For this, by opposite to traditional leisure games, they integrate different intervention mechanisms to guide patients towards the fulfillment of their goals, from instruction messages to automatic realization of part and even all the task.

The main drawback of CNVR is that the use of technology introduces a complexity factor alien to the rehabilitation process. In particular, it requires patients to acquire spatial abilities (Satalich, 1995) and navigational awareness (Chen & Stanney, 1999) in order to find ways through the VE and perceive relative distances to the patient’s avatar. In addition, interacting with virtual objects involves recognizing their shape and semantics (Nesbitt et al., 2009). Moreover, pointing, picking and putting virtual objects is difficult, especially if the environments are complex and the objects size is small (Elmqvist & Fekete, 2008). Finally, steering virtual objects through VEs needs spatio-temporal skills to update the perception of the relative distance between objects through motion (Liu et al., 2011). These skills vary from one individual to another, and they can be strongly affected by patient’s neuropsychological impairments. Therefore, it is necessary to design strategies to make technology more usable

and minimize its side effects in the rehabilitation process. These strategies, combined with customized intervention strategies, can contribute to the success of CNVR.

In this paper, we propose and discuss several strategies to ease the navigation and interaction in VEs. Our aim is to remove technological barriers as well as to facilitate therapeutic interventions in order to help patients reaching their goals. Specifically, we present mechanisms to help objects picking and placement and to ease navigation. We present the results of these strategies on users without cognitive impairments.

2. Related work

Several studies have shown that virtual objects manipulation improves the performance of visual, attention, memory and executive skills (Boot et al., 2008). This is why typical 3D CNVR systems aimed at training these functions reproduce daily life scenarios. The patients exercises consist mainly in performing virtually domestic tasks. The principal activity of patients in the virtual environments is the manipulation of virtual objects, specifically, picking, dragging and placing objects (Rose et al., 2005). These actions can be performed through a simple user interaction, in general a user click having the cursor put onto the target. More complex actions can be performed, such as breaking, cutting, folding, but in general, all can be implemented as pre-recorded animations that can also be launched with a simple user click (Tost et al., 2009).

The intervention strategies that help patients in realizing these activities by their own consist in reminding the goal through oral and written instructions and attracting the patients attention to the target through some visual mechanisms. Difficulties arisen from the use of the technology are related to the users ability in recognizing target objects, understanding the rules and limitations of virtual manipulation in comparison to real manipulation and managing the scale of the VE. In particular, the selection of small objects in cluttered VE can be difficult. To overcome this problem, a variety of mechanisms has been proposed (Balakrishnan, 2004), mainly based in scaling the target as the cursor passes in front of it. Another interesting question is the convenience of decoupling selection from vision by using the relative position of the hand to make selections, or by applying the hand-eye metaphor and computing the selected objects as those intersected by the viewing ray (Argelaguet & Andújar, 2009).

To be able to reach the objects and manipulate them, patients must navigate in the environment. However, navigation constitutes another type of activity that involves by itself a lot of cognitive skills, some of them different that those needed for objects manipulation. It requires spatial abilities, namely spatial orientation, visualization and relations (Satalich, 1995) and temporal skills to perceive the direction of movement and the relative velocity of moving objects. Moreover, navigation is closely related to way-finding. It requires not only a good navigational awareness, but also logic to perform selective searching of the target in semantically related locations, visual memory to remember the places already explored and strategy to design efficient search. In fact, although some controversy exists on if virtual navigation enhances or not real navigation abilities (Richardson et al., 2011), virtual navigation by itself is being used for the rehabilitation of spatial skills after brain damage (Koenig et al., 2009).

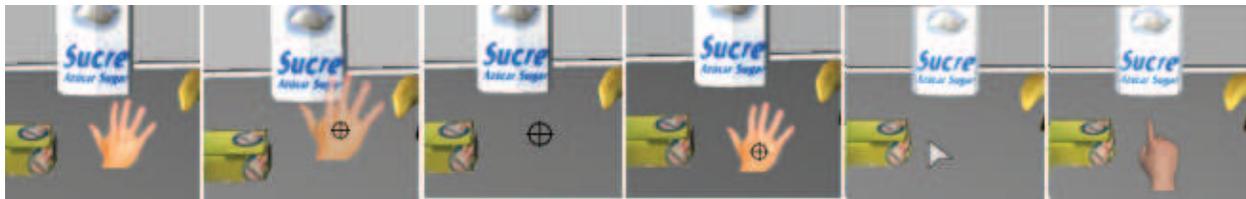


Fig. 1. The six cursors tested. From left to right: opaque hand, transparent hand + spy-hole, spy-hole, opaque hand + spy-hole, arrow, pointing finger

From a technological point of view, navigating interactively is far more complex than clicking. It involves various degrees of freedom and a non-trivial translation of the input device data into 3D motion. Therefore, in objects manipulation tasks, navigation interferes with the foreseen development of the activity. It can hinder it, delay it, and even make it impossible. In order to decouple navigation from object manipulation, automatic camera placement methods must be designed. This topic has been largely addressed by the computer graphics community (Christie & Olivier, 2008) in order to compute automatically the best camera placement for the exploration of virtual environments (Argelaguet & Andújar, 2010) and for the visualization and animation of scientific data (Bordoloi & Shen, 2005). Two main approaches have been proposed: reactive and indirect methods. Reactive methods apply autonomous robotics strategies to drive the camera from one point to the other through the shortest possible path, avoiding obstacles. They apply to the camera the navigation models that are used for the animation of autonomous non-player characters (Reese & Stout, 1999). Indirect approaches translate users needs into constraints on the camera parameters, which they intend to solve (Driel & Bidarra, 2009).

In leisure video games, camera positioning cannot be totally automatic, because camera control is usually an essential part of the game. The camera is placed automatically at the beginning of the game or at the transition between scenarios. In this case, automating positioning must preserve the continuity of the game-play, while providing the best view of the environment. A specially challenging problem is the computation of the camera position in third person games, in which the camera tracks the user's avatar. In this case, it is necessary to avoid collisions of the camera that may produce disturbing occlusions of parts of the environment (Liu et al., 2011).

The aim of our work is to design methods that reduce the technological barriers of VEs for memory, attention and executive skills rehabilitation. We extend existing techniques to ease objects manipulation, and we explore their use as intervention strategies. Moreover, in order to decouple navigation from objects manipulation, we provide automatic and semi-automatic camera placement. We show that automatic camera control provides mechanisms of intervention in the task development that ease a free-of-error training.

3. Objects manipulation

3.1 Technological assistance

We apply the eye-hand metaphor. We propose several strategies to ease pointing, picking and putting: cursor enrichment, objects outlining and free surfaces highlighting.

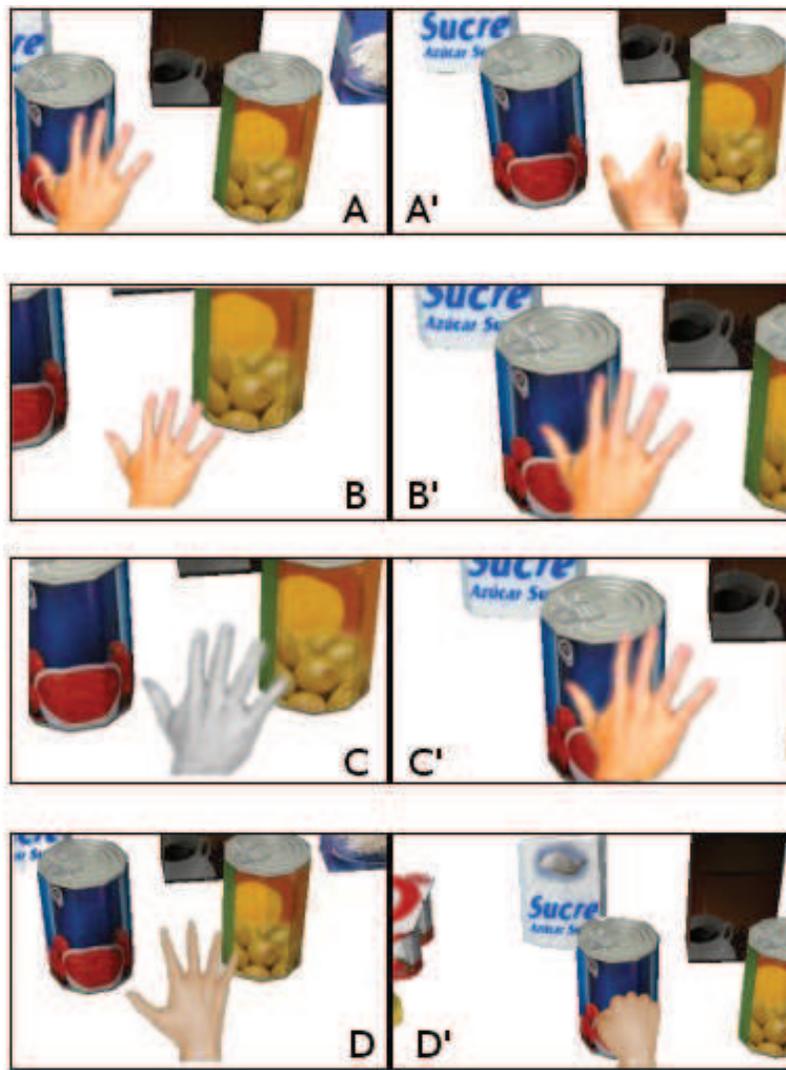


Fig. 2. The four cursor strategies to help pointing; A to A' changing shape; B to B' changing size; C to C' changing color; D to D' animating.

We have tested six different types of widgets for the cursor (see Figure 1): an opaque hand, a spy-hole, an arrow, an opaque hand with a spy-hole over-impressed, a transparent hand with a spy-hole over-impressed and a pointing finger. The hand and the finger have the advantage of helping users to understand that they are able to interact with the environment. The main drawback of the hand is that it can occlude objects. This can be corrected making it transparent. Another inconvenient is its lack of precision, which can be corrected over-impressing a spy-hole on it. The pointing finger solves this problem. The arrow has the advantage of being small and precise. However, it has a low symbolic value. The spy-hole is precise and little occlusive but it gives a non-desirable aggressive look to the task.

We have proposed two different mechanisms to help pointing: cursor-based mechanisms and object-based mechanisms. The aim of these techniques is to signal in a way or another that the object under the cursor is selectable. For the cursor, we have analyzed four different possibilities (see Figure 2): changing its shape, when it is in front of selectable objects, (ii) enlarging it, (iii) changing its color, and (iv) launching a small animation. For the

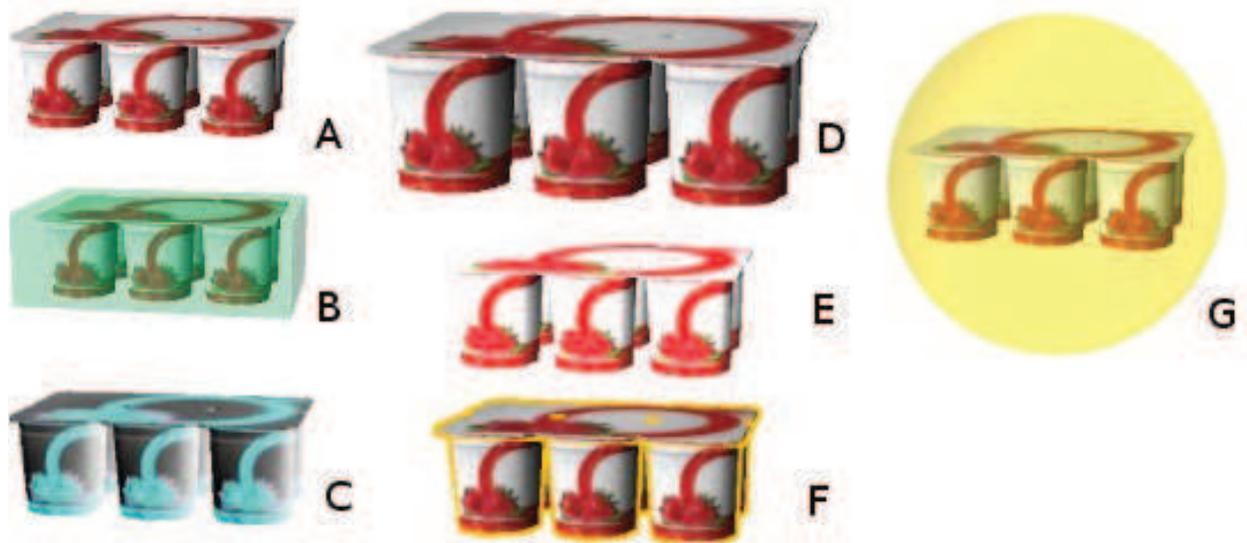


Fig. 3. The six target object strategies to help pointing; A: normal object; B: with rectangular halo; C: with complementary colors; D: enlarged; E: with increased luminance; F: with silhouette edges; G: with spherical halo.

selectable objects, we have tested 6 different ways of highlighting them (see Figure 3): (i) with bounding box halo, (ii) a circular halo, (iii) enlarging size, (iv) changing their color by their complementary color, (v) drawing silhouette edges and (vi) increasing luminance.

In a first person game with the eye-hand metaphor, dragging objects at their real scale in the VE can cause collision problems with the other elements of the scenario and occlusions in the view frustum. Therefore, instead of moving the geometric model of the object, we actually move a scaled version of the object projected onto the image plane. We have tested three different strategies: (i) to substitute the cursor by the object when it is dragged, to keep the cursor and show the dragged object (ii) centered under the cursor and (iii) separated from the cursor, at right and below it. We have tested two variants of the three strategies with and without transparency. Figure 4 illustrates these modes.

A drawback of not moving the actual object is that when users must put it down on a surface, they don't have a good spatial perception of the free space left. Therefore, we highlight the free space able to lodge the held object as users move the cursor. We have tested different ways of highlighting the surface applying different colors and drawing the 3D bounding box that the object would occupy in the surface or only its projected area (see Figure 5).

3.2 Cognitive assistance

Technological aids can also be used to assist patients at the cognitive level. In particular, to assist patients in picking a specific object, instead of outlining all the pickable objects, we can outline only those related to the task goal. In this case, outlining fulfills two different functions: technological aid and cognitive assistance. The number of visual stimuli is reduced to only those that are related to the task. As a consequence, the range of strategies to outline objects is larger: in addition of appearance changes, we can apply sounds and animations that are not suitable when there is a large number of objects to be outlined.

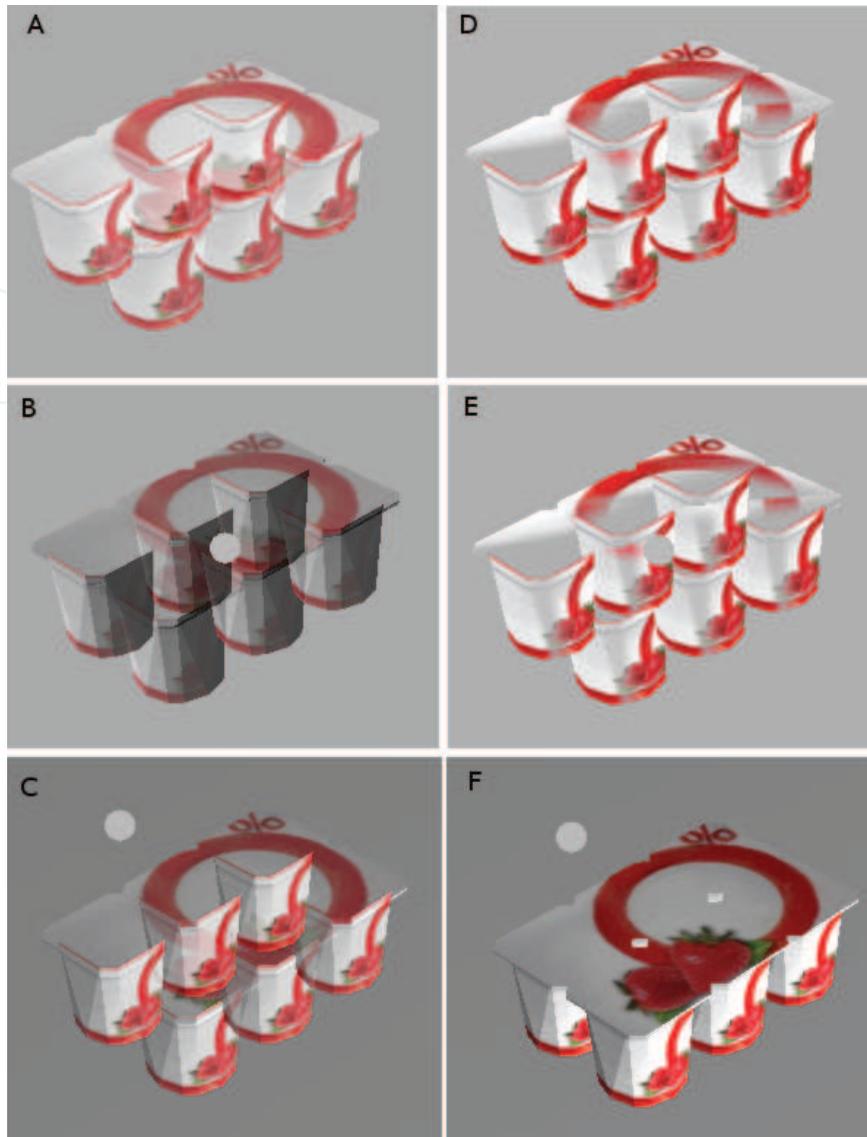


Fig. 4. Different strategies to give feedback of the dragged object: left column transparent feedback, right column opaque feedback; first row the cursor is substituted by the object; second row, the cursor is centered on the object; third row, the object is at bottom right of the cursor.

Cognitive assistance can also be provided through game-master actions. The game-master is a component of the game logics that simulates the intervention of an external observer. It helps the patient by emitting instructions and feedback messages, removing objects of the scenario to simplify it, demonstrating the required action or doing it automatically partially or totally. This way, if focus of the task is put on picking objects and not placing them anywhere else, the picking action can be implemented as a pick-and-place: users pick and the game-master places them.

4. Navigation methods

We propose four different modes of navigation: free navigation, two user-assisted navigation modes and a fully automatic navigation.

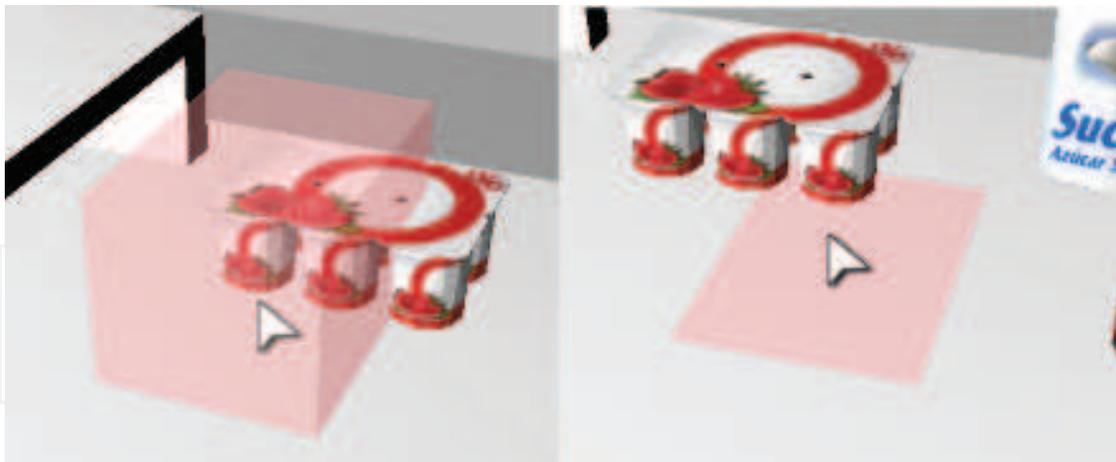


Fig. 5. Two different strategies of highlighting the space that a dragged object would occupy on a surface: at right the 3D bounding box, at left the projected area.

4.1 Free navigation

The free navigation is the classical first person four degrees of freedom model. Users control the viewing vector orientation through the yaw and pitch angle. As usual in computer games, rolling is not allowed. The pitch angle is restricted within a parameterized range between -50 to 50 degrees to allow looking at the floor and ceiling but forbidding complete turns. In addition, users control the camera position by allowing its movement in a plane parallel to the floor at a fixed height. Jumping and crouching down are not allowed. The movement follows the direction of the projection of the viewing vector in that plane, therefore it is not possible to go back. Users can also stop and restart the camera movement. The movement has constant speed except for a short acceleration at its beginning and deceleration at its end. The camera control is done using the mouse to specify the viewing vector and camera path orientation and the bar-space key to start and stop the motion. This system has the advantage that it requires to control only two types of input (mouse movement and space bar key), which is suitable for patients with neuropsychological impairments.

4.2 Assisted navigation

The aim of the assisted navigation mode is to provide means for users to indicate where they want to go, and then automatically drive them to this location. This way, the focus is put on the destination and not on the path towards it. Therefore, navigation is decoupled from interaction.

This assistance 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 indicate the target 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 petition of interaction (to open, pick, put or transform),

and it realizes it according to the task logics. However, if the object is not reachable, the system interprets that the user wants to go towards it, it computes the corresponding path and follows it automatically. To provide more user control on the navigation, the system allows users to stop and restart the navigation at any time. Observe that the target location may not be directly the object that the user needs to manipulate, but a container in which the object is hidden. For instance, if the goal of the task is to take out a chicken from the oven, the target direction is the oven's door in order to open it. Depending on the current level of difficulty of the task user will have more or less precise instructions on their goal. This is precisely one of the objectives of the rehabilitation.

The main difficulty with this strategy is that clicking onto the target requires to have it in the view frustum and to put the cursor onto it. However, both things require a previous navigation or camera orientation process. We distinguish two cases: (i) when the target can be seen without need of modifying the camera position, but only its orientation, and (ii) when the camera position must be modified. In the former case, we propose two strategies: *free camera orientation* and *restricted camera orientation*. The *free camera orientation mode* has two degrees of freedom: yaw and pitch. The camera position is fixed, and users move the viewing direction until the target is in the center of the view frustum. The *restricted camera orientation mode* has one degree of freedom. The system performs an automatic rotation of the yaw angle. Users only modify the pitch angle. To select the desired orientation, users stop the rotation with a mouse click.

When the target location is invisible from the current camera 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. In this case, although navigation is removed, way-finding cannot be eliminated. Therapists must be aware of that in the design of their tasks. To overcome this problem, inside rooms, we design the scenarios avoiding the presence of occluders. When the scenario is composed of various rooms, we avoid corridors, we design doors from one room to the other and put the name of the room on the doors. This way, to indicate the direction to another room, users click onto the corresponding door.

4.3 Automatic navigation

The automatic navigation method removes the camera control from users. The system computes the target destination according to the task logics. It puts the camera in front on the next object with which users must interact. For instance, if users are asked to pick a tomato, the application places the camera in front of one.

With this mode, the system intervenes in the task development: it takes decisions on the places to go, and therefore eases the task. It is part of the possible intervention strategies that therapists can design to help their patients.

4.4 Implementation

In order to manage the described alternatives, the system needs to know the position of all interactive objects. The position of static objects is part of the scene model. Objects that can be moved can be on top of or inside other objects. We use a system of grids that allows us to control the exact position of all the objects at any time of the play. Then, when an object is the

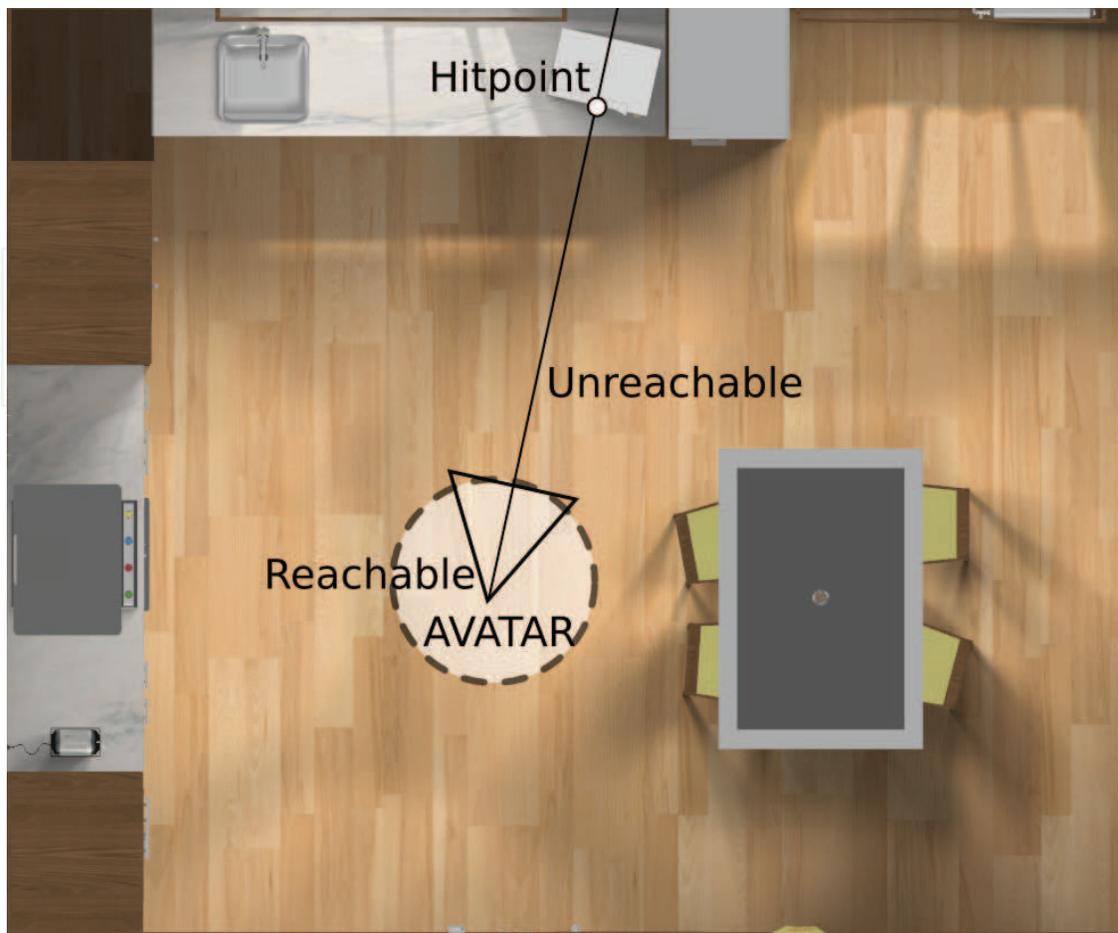


Fig. 6. VE example that shows the avatar's position and the first intersection of the view-vector with the scene. In this case the hit-point is the microwave, and it is not reachable for interaction at the current position.

user's target, the system has the ability to compute the best position to reach it, computing the best path to arrive there and the best camera's orientation.

When the user does a mouse click, the system detects the first object that intersects the view vector (hit-point). Figure 6 shows an example where the hit-point is a microwave. The object is reachable if the distance between its position and the avatar is smaller than a fixed value. In this case, the interaction with the object is performed. When the distance is greater, the system interprets that the user wants to perform the interaction, and it moves the avatar to a position that allows the interaction with the object.

The system uses the object's position to determine the best avatar's location to reach it. This location lays on the grid of the VE's floor. Figure 7 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

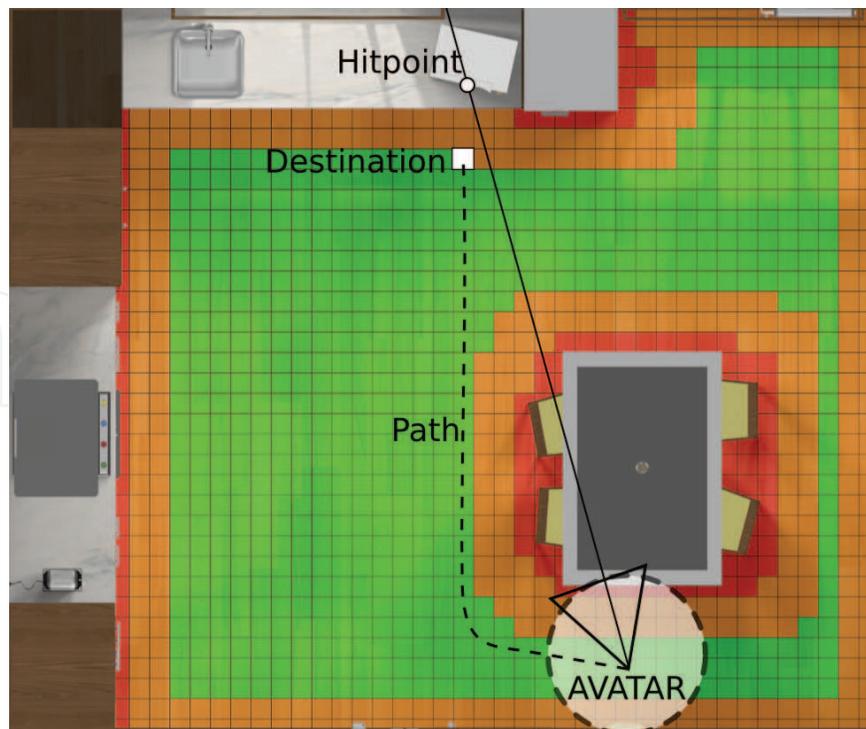


Fig. 7. Example of the navigation strategy. The system use the floor's grid to determine the best path to reach the hit-point. The destination cell is draw in white, and the cells are classified as occupied (red), unreachable (orange) and reachable (green).

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 the 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 Bezier path that follows the discrete path computed before. This new path allows the system to perform softer movements and keep a constant speed.

5. Results and discussion

In order to test the suitability of the proposed technological assistance strategies, we have created a set of specific tasks in a virtual scenario representing a kitchen. We have asked 30 volunteers without cognitive impairments to realize these tasks. We have recorded their results and asked them to fill a questionnaire about different aspects of these assistance strategies. The profile of the users can be categorized according to three different criteria:



Fig. 8. A task example to test the suitability of the different types of cursors.

gender, age (below and over 30) and 3D-game players or not. The groups were approximately of the same size except for women over 30 and game-player and men below 30 non 3D game-players that were smaller (only 1 and 2 users, respectively).

5.1 Object manipulation tests

To test the strategies to help picking, we put the camera in front of an open virtual fridge full of objects. The camera movement was inhibited. User could only rotate the camera but not change its position. All the objects were accessible from the camera position. The task consisted of clicking onto objects to remove them. Figure 8 shows an example of this task. We measured the number of objects that users could remove during a fixed time interval. Users answered questions about their preferred mechanism according to three different criteria: visibility, precision and visual appearance. We tested the different cursors as well as the objects highlighting mechanisms. The results showed that the cursor that allowed a larger number of selections was the spy-hole. However, users preferred the pointing finger because they found it more meaningful. Their second preference was the arrow. In addition, users preferred the animation of the cursor to indicate the nature of the interaction that objects support, for instance, a rotation of the hand on doors, and opening and closing the hand for grasping objects.

To test objects grabbing, we set a task in which users had to move objects from the fridge to the kitchen marble at a side. The marble was also reachable, so it was not necessary to move the camera but only to rotate it. We measured the number of objects that users were able to move



Fig. 9. The three cursor modes: at left an arrow on unreachable objects, in the middle an animation of the hand on reachable objects and, for dragging, the arrow and the held object conveniently scaled, here, the knife.

during a fixed time interval. Users answered questions about visibility, precision and visual appearance. The results showed the more effective mechanism was to show the grasped object at a side of the cursor. Surprisingly, the opaque mode was preferred to the semi transparent one, because it was found more natural.

Taking into account these results, we finally chose to have three cursor modes (see Figure 9): when the object under the cursor is not reachable or an object is being held, we use an arrow and when the object is reachable the animated hand.

Finally, to test the feedback mechanisms to help putting objects, we used a task consisting in placing objects on the kitchen marble. Users did not need to pick them. As soon as they put one, another was automatically grasped. We measured the number of objects that users were able to leave on the marble, and users answered a questionnaire about precision and comfort. The preferred mechanism was to color the 2D free space nearer the cursor.

5.2 Navigation methods test

To test the navigation methods we proposed a task consisting in touching two objects strategically placed in the scenario (see Figure 10): an orange (dashed in blue) and a dish (dashed in pink). The task is segmented in three stages. The first stage is to reach the orange. In this stage, it is necessary to avoid the table, which is an obstacle that does not prevent from seeing the target object. The next stage is to go to the cabinet's door, which consists of walking in a straight line near the marble without obstacles. The last stage is to open the cabinet door and walk around to reach the dish, which is an object placed into a container. Thus, the cabinet door is an obstacle that prevents from seeing the target object. Figure 11 represents the paths performed by the users in free navigation mode using a temperature range color encoding. Temperatures are represented with a blue-to-red color scale. The most transited is the place that represents the higher temperature. In other words, places where users stay during a long time are colored in red.

Clearly, it can be seen that areas containing obstacles (near the table in the first stage and near the door in the third stage) are critical for users. The area corresponding to the second stage is also colored in red because when there are no obstacles, they all choose the same



Fig. 10. Results of the paths used by the users to perform the task.

option: to walk in a straight line near the marble. The black line represents the automatic path computed by the system. Our method keeps constant speed along the path and avoids the obstacles automatically without errors. Note that some of users chose the other side of the table which is the longest way. These users had, in general, more difficulties in navigating, as it can be seen from their paths sparser and more erratic.

The temperature diagram in Figure 11 represents the head deviation in the pitch angle in free navigation mode. The figure shows that users tend to look down to see where they are going.

Users ranked the navigation method easiness in the following order: automatic, assisted, assisted rotation and free. As expected, they found the automatic method very easy or easy (100% of users). The assisted mode was also considered as easy and very easy by 95% of users. However, the assisted rotation was less valued (only 80% scored it as easy or very easy). Users reported that not being able to rotate the camera was disturbing. The sensation was the same in all groups. The free navigation mode was found easy and very easy by 67% of users in general, but only by 20% of non 3D-gamers, who were, however, 2D-gamers. We conclude, as expected, that free navigation is difficult for non-gamers. Concerning the preferred navigation mode, all groups of users chose the assisted mode, even the 60% of game players.

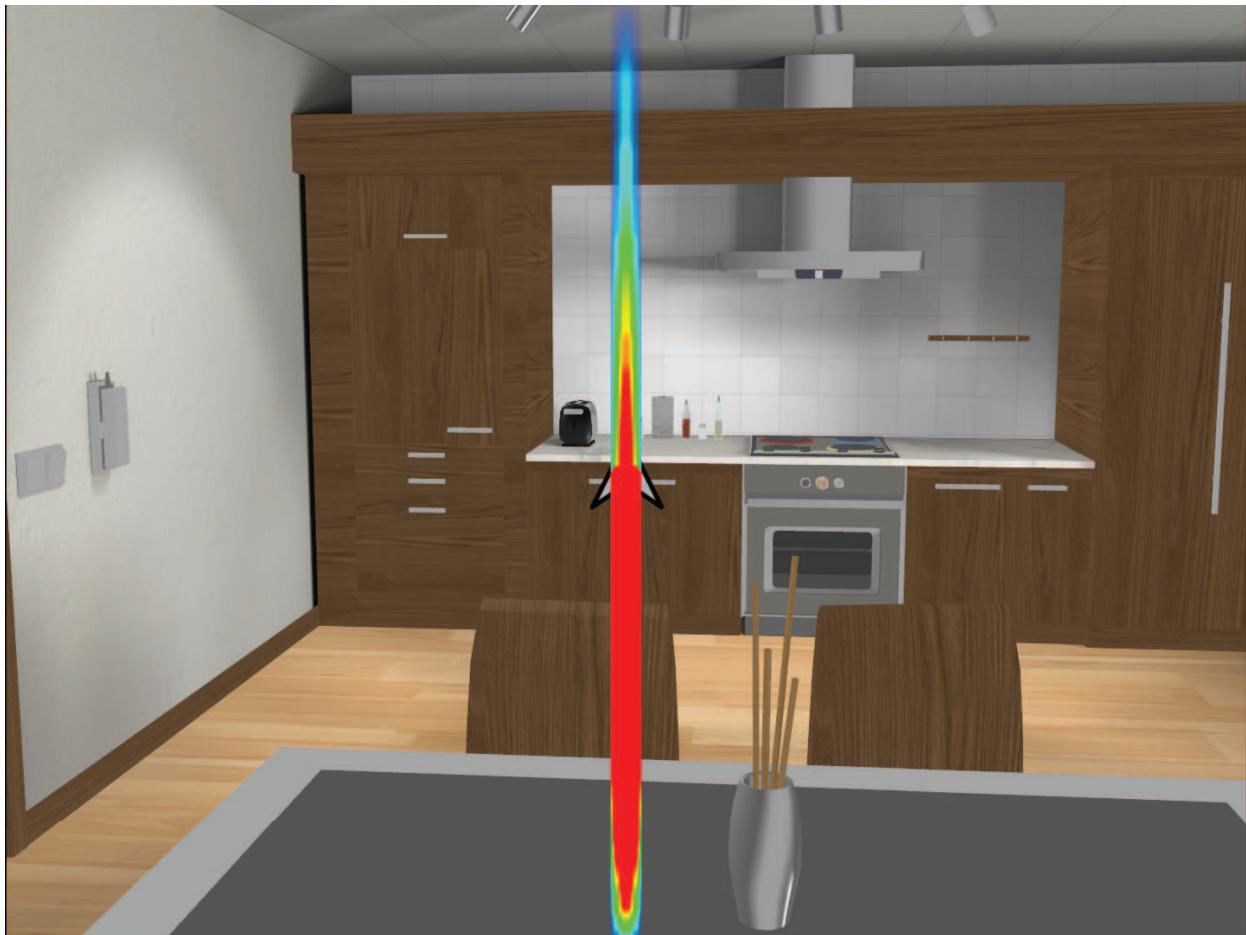


Fig. 11. Results of the camera movements performed by the users to encountering the task.

In relation to the quality of the paths, 90% of users valued them as good and very good. About 75% of users described the camera orientation during movement as good and very good whereas 25% found it regular or bad. The bad scores came from users that reported being disturbed by the fact that the camera didn't look at the target when it was avoiding an obstacle. They affirmed that the target object should always be in the view. For this reason, we modified the system to allow users to control the camera rotation if they wish during the automatic camera movement. In this way, the system is more flexible and satisfies the desires of passive users as well as more active ones.

6. Conclusions

Serious games in 3D virtual environments can greatly contribute to rehabilitation. However, to be usable by patients with cognitive impairments, many technological barriers must be broken that are disturbing for all kind of users as well. In particular, it is important to separate objects manipulation from navigation. In this paper, we have proposed and analyzed several strategies to ease manipulation in 3D. We have designed visual mechanisms to enhance the perception of objects. Our aim is to help users in picking, dragging and putting objects. In addition, we have proposed a mechanism for the semi-automatic navigation inside VEs. Our goal is to allow users to move in the VE without having the technological difficulties of

controlling a virtual camera. We have tested these strategies with volunteer users and seen that they were effective and well accepted. Our next step is to use them on patients.

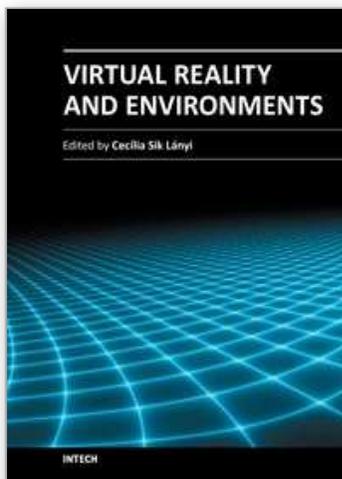
7. References

- Argelaguet F. & Andújar C. (2009). Efficient 3D pointing selection in cluttered virtual environments, *IEEE Computer Graphics and Applications* 29(6): 34–43.
- Argelaguet F. & Andújar C. (2010). Automatic speed graph generation for predefined camera paths, *Smart graphics*, pp. 115–126.
- Balakrishnan R. (2004). "Beating" Fitts' law: virtual enhancements for pointing facilitation, *International Journal on Human-Computer Studies* 61: 857–874.
- Boot W., Kramer A., Dimons D., Fabiani M. & Gratton G. (2008). The effects of video game playing on attention, memory, and executive control, *Acta Psychologica* 129(3): 387–398.
- Bordoloi U. & Shen H.-W. (2005). View selection for volume rendering, *IEEE Visualization'05*, pp. 487–494.
- Chen J. L. & Stanney K. M. (1999). A theoretical model of wayfinding in virtual environments: Proposed strategies for navigational aiding, *Presence* 8: 671–685.
- Christie M. & Olivier P. (2008). Camera control in computer graphics, *Computer Graphics Forum* 27(8): 1–197.
- Driel L. & Bidarra R. (2009). A semantic navigation model for video games, *Proceedings of the 2nd International Workshop on Motion in Games, MIG'09*, Springer-Verlag, pp. 146–157.
- Elmqvist N. & Fekete J. (2008). Semantic pointing for object picking in complex 3D environments, *Proceedings of Graphics Interface 2008, GI '08*, pp. 243–250.
- Guo W. Lim S. Fok G. & Chan, G. (2004). Virtual reality for memory rehabilitation, *International Journal of Computer Applications in Technology* 21(1–2): 32–37.
- Koenig S., Crucian G., Dalrymple-Alford J. & Dunser A. (2009). Virtual reality rehabilitation of spatial abilities after brain damage, *Studies for Health Technological Information* 144: 105–107.
- Liu L., Jean-Bernard J. & Van Liere R. (2011). Revisiting path steering for 3D manipulation tasks, *International Journal on Human-Computer Studies* 69: 170–181.
- Nesbitt K., Sutton K., Wilson J. & Hookham G. (2009). Improving player spatial abilities for 3D challenges, *Proceedings of the 6th Australasian Conference on Interactive Entertainment*, pp. 1–6.
- Prigatano G. (1997). Learning from our successes and failures: Reflections and comments, *International Neuropsychological Society* 3: 497–499.
- Reese B. & Stout B. (1999). Finding a pathfinder, *Symposium on artificial intelligence and computer games*, pp. 69–72.
- Richardson A., Powers M. & Bousquet L. (2011). Video game experience predicts virtual, but not real navigation performance, *Comput. Hum. Behav.* 27: 552–560.
- Rose F., Brooks B. & Rizzo A. (2005). Virtual reality in brain damage rehabilitation: Review, *CyberPsychology & Behavior* 8(3): 241–271.
- Satalich G. (1995). Navigation and wayfinding in virtual reality: Finding the proper tools and cues to enhance navigation awareness, [Http://www.hitl.washington.edu/publications/satalich/](http://www.hitl.washington.edu/publications/satalich/).

Tost D., Ferré M., Garcia P., Tormos J., Garcia A., Roig T. & Grau, S. (2009). Previrnec: a cognitive telerehabilitation system based on virtual environments, *Virtual Rehabilitation*, pp. 1–8.

IntechOpen

IntechOpen



Virtual Reality and Environments

Edited by Dr. Cecília Sík Lányi

ISBN 978-953-51-0579-4

Hard cover, 204 pages

Publisher InTech

Published online 27, April, 2012

Published in print edition April, 2012

Virtual Reality is clearly interdisciplinary research. It has, not only Information Technology importance but social, educational, economical importance too. It combines multiple disciplines for the development of virtual reality systems in which the user has the immersive feeling of being in the real world. Virtual reality has several applications in almost all fields of real life. The most typical fields for the application of virtual reality are health-care, engineering and game industry. This book may be a solid basis for the novice and advanced engineers who would like to develop user friendly Virtual Environments for education, rehabilitation and other applications of Virtual Reality. Our book provides a resource for wide variety of people including academicians, designers, developers, educators, engineers, practitioners, researchers, and graduate students.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Dani Tost, Sergi Grau and Sergio Moya (2012). Personalization of Virtual Environments Navigation and Tasks for Neurorehabilitation, Virtual Reality and Environments, Dr. Cecília Sík Lányi (Ed.), ISBN: 978-953-51-0579-4, InTech, Available from: <http://www.intechopen.com/books/virtual-reality-and-environments/personalization-of-virtual-environments-navigation-and-tasks-for-neurorehabilitation>

INTECH
open science | open minds

InTech Europe

University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China

Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

© 2012 The Author(s). Licensee IntechOpen. This is an open access article distributed under the terms of the [Creative Commons Attribution 3.0 License](#), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

IntechOpen

IntechOpen