AN EXPERIMENTAL COMPARISON OF THREE METHODS FOR COLLISION HANDLING IN VIRTUAL ENVIRONMENTS

Jeffrey Jacobson & Michael Lewis
goshen@cmu.edu ml@sis.pitt.edu
Dept. of Information Science, University of Pittsburgh.
135, Bellefield Ave, Pittsburgh, PA, 15213 USA
+1 412 624 9468

This study compares three common strategies for handling collisions between the user's virtual body and other objects in a cluttered virtual environment. Test subjects sought "treasures" in a maze of narrow corridors which were embedded in a jumble of irrelevant shapes. The application ran on a PC, with the mouse and screen as the interface. When encounters an object, he either passes through it, stops completely, or is deflected around it. Data show that the third strategy best facilitates goal-seeking behavior with this interface and for this type of problem. This result is significant because collision handling is critically important to the usefulness of Virtual Reality applications. Furthermore, the screen-and-mouse interface is both the most common and least studied for virtual environments.

INTRODUCTION

This study shows that collision handling has a strong effect on whether the user can move able to move between and around objects smoothly in a virtual environment. In our observations, existing collision strategies fall into three categories, which we term:

- **Ghost** User passes through object. Used in architectural walkthroughs (Hudson, 1997) and simpler virtual reality implementations. (Marrin and Campbell, 1997)
- **Clunk** Complete stop on contact with object. Seen in the old tank games. (Battlezone, 1980)
- **Slip** User's movement is deflected making him slide around the object. Used in the best "first-person view" video games (Quake, 1997) and advanced graphics libraries. (IRIS Performer, 1997)

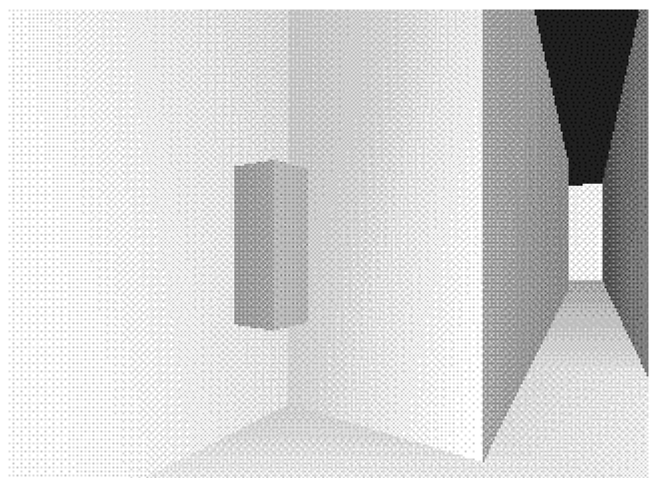


Fig 1: A view of the inside of the maze. The floating rectangular object is a treasure.

Our study shows that slip mode provides the user with the efficient means of navigation, in a reasonable virtual environment. The superiority of slip mode was shown in the greater speed subjects using it were able to locate objects in a maze. The maze consisted of narrow corridors, low-lying obstacles and extra baffles between walls. (Fig 1 & 2)

In this paper we present:

- The maze, the objective (treasure hunting) and the collision modes are representative of elements found in actual VR applications.
- The software implementation based on the explicit use of human "personal space."
- Task timing results showing the superiority of slip mode.
- Scatterplots showing where test subjects actually spent their time in the maze.
- How the results generalize to practical applications.

MOTIVATION

To navigate a cluttered environment, one must be able to plan a route without considering every step. In the real world, we use low-level psychomotor behaviors to get around objects. (Turvey and Kugler, 1987) For example, suppose someone is in a room and moving toward the door, but there is chair in the way. Rather than plan some perfect route around the chair, she will most likely move straight toward the door, and step around the chair when she gets to it; all without much thought.

Unfortunately, most VR interfaces available today provide neither the range of perception nor the body control for the user to employ these reactive behaviors. The problem is especially acute with the most commonly used physical computer interface: the screen-and-mouse. The collision strategies we saw generally fell into the three categories, which we term, "ghost", "clunk" and "slip."

Ghost Mode

Ghost mode is actually the absence of collision handling--one simply floats through objects and walls. In a virtual environment, ghost mode reduced or eliminates the need to avoid objects, but limits the types of applications one can sensibly build. For example, ghost mode might be perfect for exploring some otherwise closed structure, like the human body. However, it would be inappropriate for many military training applications.

Operationally, the ghost mode user can also get lost in places where the visual field is cluttered, as with most locations in the maze. Even under the best of circumstances, ghost users would have to learn both sides of every wall to fully understand the maze. Added baffles between the walls make it more confusing, (fig 1) but this is representative of the topologies one might find in simulated office buildings, nuclear power plants or complex machinery.

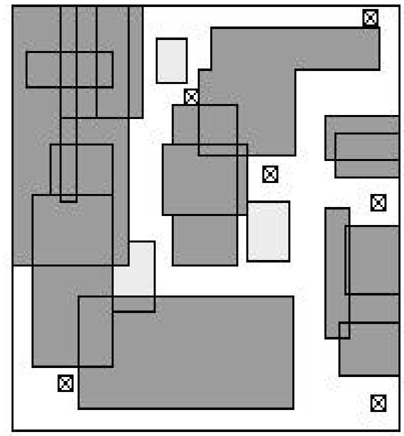


Fig 2: Map of the maze. The square symbols are treasures.

Clunk Mode

In clunk mode, the user simply stops when he encounters an object. In uncluttered environments (Battlezone, 1980) (Sperger, 1996) this works reasonably well, but not in spaces that are narrow or cluttered. There, the merest brush with any object causes the user to come to a full stop. Before he can start moving again, he must back up and reorient. Sometimes, objects are out of the user's field of view, which in most applications is only about 100 degrees, compared to almost 200 in the real world. In the act of moving to free himself, the user may touch some other objects and be stopped again. Clunk mode is impractical for many situations; even a virtual living room can become unmanageable.

The walls in the maze were deliberately made narrow with respect to the user's virtual body (Fig 1) and some low-lying obstacles were put in place, to characterize difficulties users face in many applications.

Slip Mode

Upon collision, the user's movement is deflected, effectively making him slide around the object he ran into. Slip mode appears to be the best compromise for "realistic" situations, preserving the solidity of objects while allowing comfortable movement. Slip mode simulates the overall effect of the reactive behaviors people use in reality to get past obstacles.

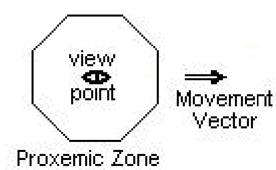


Figure Three: Proxemic Zone. Overhead view, shown here with only eight sides for clarity.

Objectives

We compare the desirability of the three navigation modes. An indirect measure of this is the relative efficiency with which of subjects can find treasures in the maze, using different collision modes. We created a simple walkthrough application, the maze itself, and a representative implementation of each of the collision handling modes.

IMPLEMENTATION

The virtual body is implemented as a single viewpoint, but it is surrounded with an invisible cylindrical cage which represents the user's personal space. We call it the *proxemic zone*, and it is implemented as a symmetric sixteen-sided prism that extends from below the user's field of view to just above it.

In **ghost** mode, the viewpoint is moved according to mouse input without regard to collisions.

In **clunk** mode, the proxemic zone moves ahead of the viewpoint in the direction the mouse indicates. If no collision, the viewpoint is moved forward to the center of the zone. However, if the zone intersects any object, it is moved back to where it was and the movement vector is discarded. (fig 4)

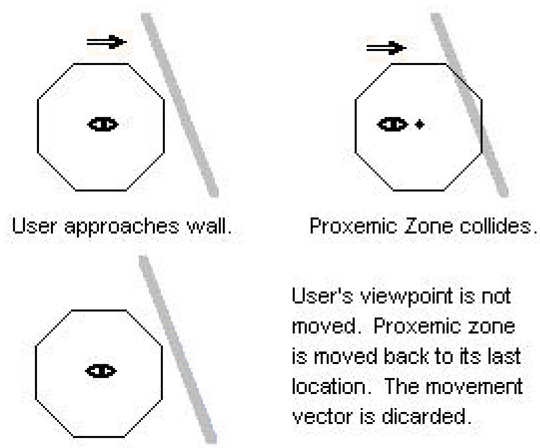


Fig 4: Implementation of clunk mode. The proxemic zone had sixteen sides, but is shown here with eight sides for clarity.

In **slip** mode, when the user's proxemic zone intersects one or more objects, the user's movement vector is minimally redirected. It is adjusted only just enough to prevent the collision from recurring in the next frame. (Fig 5)

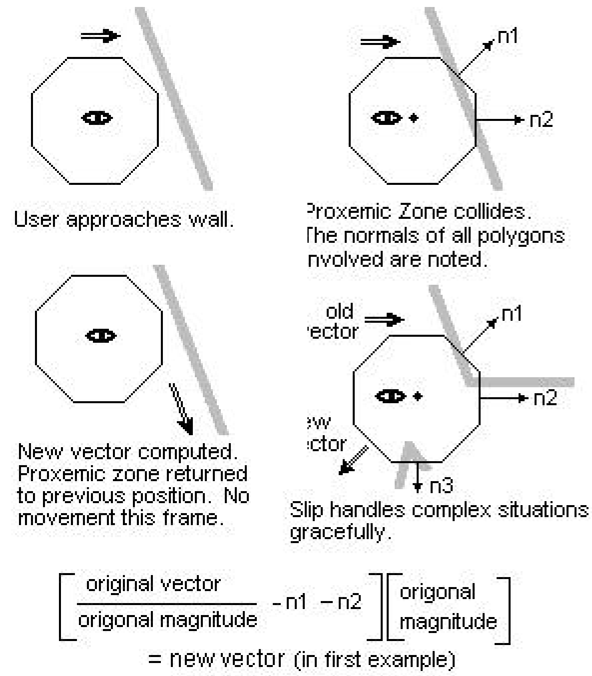


Fig 5: Implementation of slip mode.

METHOD

The application ran on a PC, in a university laboratory. Twenty-six paid subjects were recruited from the University of Pittsburgh and Carnegie Mellon communities. Subjects were randomly assigned to three groups, one for each collision handling mode. (Campbell and Stanley, 1963) Performance was measured by the average time it took each subject to get from one treasure to another. This time excludes time spent before getting the first treasure (orientation effects) and after getting the last one (diligence effects).

Upon the subject's arrival, the tester read these instructions.

> In this test, you will travel through a three-dimensional maze. You navigate with the mouse, by holding down the left mouse button. Your movement will depend on where the cursor is.
>
> - If the cursor is near the top of the screen, you will move forward. *demonstrate*
> - If the cursor is near the bottom of the screen, you will move backward. *demonstrate*
> - If the cursor is near the right side of the screen, you will ROTATE right. *demonstrate*
> - If the cursor is near the left side of the screen, you will ROTATE left. *demonstrate*
> - The closer the cursor is to the center of the screen, the slower you will go, the nearer to the edge, the faster you will go. *demonstrate and move forward*

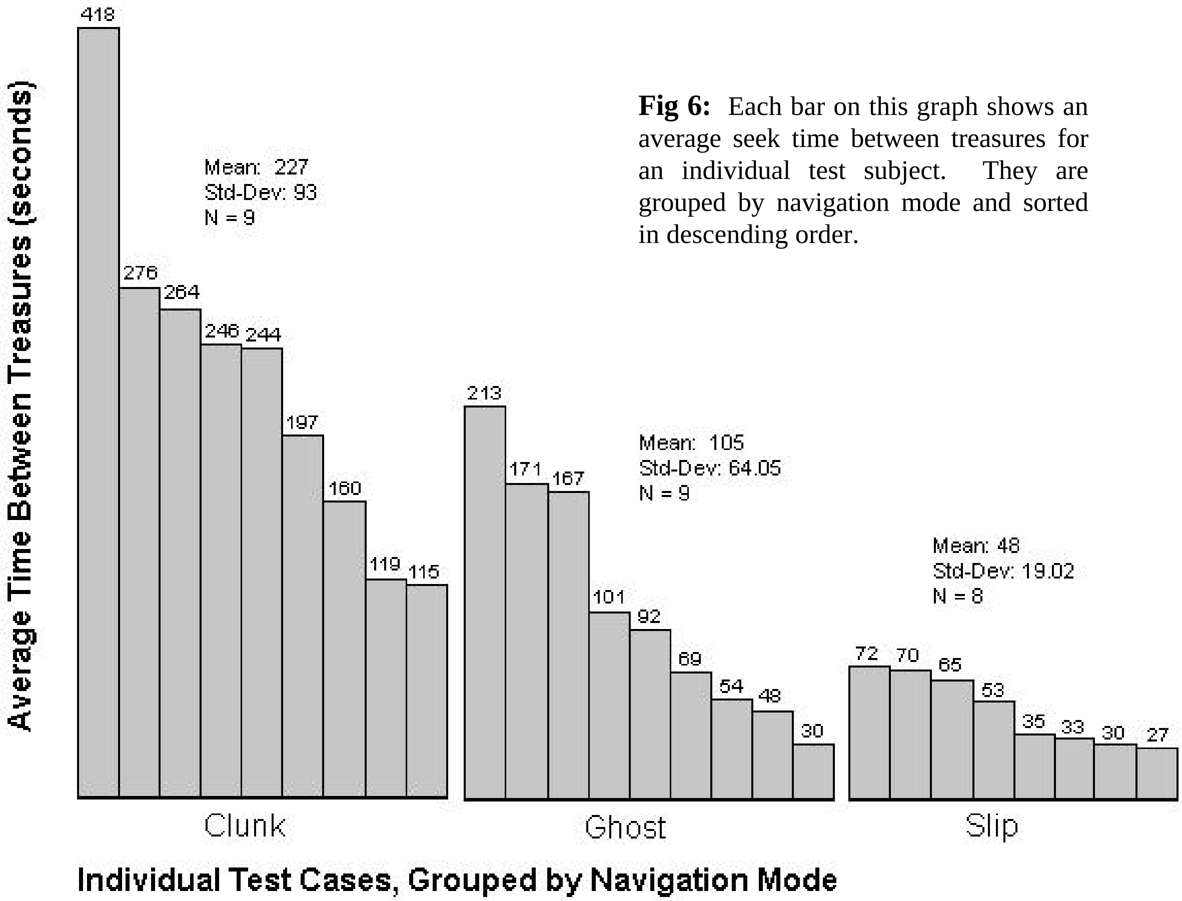


Fig 6: Each bar on this graph shows an average seek time between treasures for an individual test subject. They are grouped by navigation mode and sorted in descending order.

- If you are ambitious, you can click on a corner of the screen, which will move and rotate you at the same time. *demonstrate*

The goal is to pick up as many treasures as you can. The floating purple box in front of us is a treasure. You "get" them by moving into or onto them, like so. *demonstrate The treasure disappears when the viewpoint nears it.*

The blue up there is the sky. *Point to it.*

When you think you got all the treasures, press the space bar to quit. There is no right and wrong. You can hurry or go slowly as you choose. You are not obligated to go longer than thirty minutes, though you can if you wish.

During demonstration, the tester was careful to keep his viewpoint away from the walls and to make no mention of collisions or collision handling.

The test subject was then given the chair in front of the computer and the application was restarted. Once he appeared to be properly oriented and using the program correctly, the tester left the room and waited for the subject to tell him when he was done. During the test, software recorded when each treasure was taken, the complete path taken by the user, and other measures.

RESULTS

Generally, we found that slip mode allowed users to get treasures faster and by covering less ground.

Figure six (fig 6) shows the average search time between treasures for each of the twenty-six subjects. For each collision mode an average was computed from the search times for all subjects using it. Analysis of variance test on the three averages showed a significant difference ($p < 0.01$). (Fig 6) Also, measures of the average speed of the user's viewpoint movement indicate that Clunk users moved more slowly than the others while ghost users moved the fastest. ($p < 0.05$)

In figure seven (fig 7), the plot on the far right shows the outlines of the maze itself emerging from the pattern of the slip users' x,y locations, or footprints. The footprint density is roughly the same in most areas, which shows that slip users were generally able to go wherever they wanted within the maze. The plot for the clunk mode users looked similar, but with lower density. The plot for the ghost mode users (center of Fig 7) had to be truncated to fit in the figure. There was actually a great deal of activity outside of the maze model, sometimes far from it.

.In figure six (fig 6), compare the individual ghost users' times with the slippery users' times. Note that highly adept ghost mode users were able to get around the maze with competitive times, so a highly skilled person can use ghost mode effectively.

Most of those subjects who did do well tended to avoid going through walls, but they did cut corners, which you can see in the scatterplot in figure seven (fig 7) by the rounding of the paths near the corners of the maze. This region of tended to trap and confuse them. Also, areas of higher density correspond to open areas between the baffles. It shows that users tended to avoid walls, which makes sense. Being close to a wall uncomfortably limits the user's field of view.

DISCUSSION

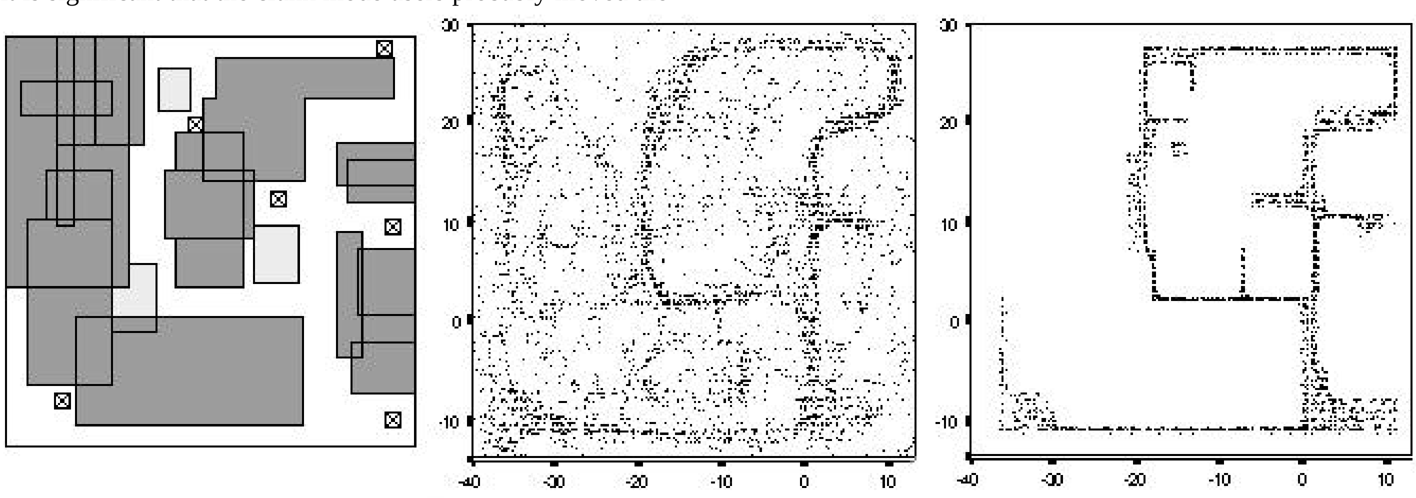
It is important to keep in mind that that the collision mode used in the application needs to be consistent with the narrative theme of the virtual environment. For example, if the user has to navigate a space which is visually open, but still has a lot of obstacles, ghost mode might be better. For ghost mode, the main problem is the potential for sudden and complete changes in the scene, which would rarely occur in an uncluttered space. Clunk mode is used in various tank games, where the landscape is also quite open but with only a few obstacles. (Battlezone, 1980) It makes sense in this case, because an actual tank really would come to a full stop upon collision with a large object. Finally, slip mode works well in a maze, but would not make sense for a flight simulator. Slip mode is best when the user is "a human" moving about a naturalistic settings.

Some ghost mode users pursued a strategy of circling outside of the maze and making short trips into it, though the walls. Once they learned the overall shape of the exterior of the maze model, this guaranteed that they did not get too lost. However, it was costly, both in time and ground traversed.

Given their much greater speed, longer seek times, and tangled paths, the ghost mode users probably moved the mouse more than the slip mode users did. With this in mind, it is significant that the clunk mode users probably moved the mouse around more than all of the others in their efforts to break free of collision conditions. That would suggest an inverse relationship between mouse motion and seek times. Unfortunately, the amount of mouse motion was not recorded during the experiments, so there is no way to tell at this point. It will have to be seen in a later study.

The delay that clunk mode users experienced came simply from the time it takes to back up, reorient, and get going again. Practice appeared to be important as can be seen by the widely different seek times. The two or three subjects who performed best seemed to know how to restart after a collision in a way that was less likely to result in another collision. The rest tended to simply jigger the mouse back and forth until they could get going again. The one very long seek time (418) (fig 6) was the result of a unique (with respect to the others) navigation strategy. The test subject was extremely patient and calculated every step. If this result is treated as an outlier and removed from the dataset, however, tests still show the mean seek time is significantly different ($p < 0.05$) from the ghost users' mean seek time.

It should be noted that because the mouse interface is first-order, moving along a straight-away does not require moving the mouse. One simply holds the cursor in its current position and continues to move. Because of their unrestricted movement, the ghost mode users probably had the lowest mouse-moving-to-footstep ratio. However, their mean performance time was still worse than that of the slip mode users. It would seem that mouse effort is a weaker determinant of performance than the collision handling strategy.



Map of the Maze   Ghost Users   Slip Users

Fig 7: On the left, is an overhead map of the original maze. In the middle is a scatterplot of 10% of the x,y location of all the ghost mode users. On the right, is a scatterplot of all locations for the slip mode users; the plot for the clunk mode users looks much the same, although the density is lower. The actual maze is more rectangular, than shown in this figure, having an aspect ratio of approximately 6/7. The figure being shown more square for layout reasons does not affect the central facts it illustrates.

CONCLUSION

These results support our thesis that slippery mode is an efficient collision resolution strategy. The maze was designed to be difficult to traverse, but all of its features are typical of what is found in existing virtual environments. The independent development of similar strategies in video games and animation libraries support the generalizability of our findings.

The utility of collision handling schemes for virtual reality warrants more detailed study. In situations such as desktop virtual reality where interaction techniques are some what arbitrary, it is especially important to identify the good ones. Slip mode does not attempt to replicate natural interactions, but instead minimizes the delay imposed by obstacles. We would like to see the development of more such pseudo-realistic interaction techniques for navigation and manipulation. They preserve the intuitions and affordances offered by virtual reality, but allow for a simple interface.

REFERENCES

1. Battlezone. Original game by Atari Inc., 1980. The PC version is available at http:// www.microsoft.com/games/arcade/prodinfo.htm#ZONE. Atari version: ATARI, INC., Consumer Division, 1312 Crossman Ave., P.O. Box 61657. (800) 538-8543 (in USA) Sunnyvale, CA 94086
2. Campbell, Donald T & Stanley, Julian C. *Experimental and Quasi-Experimental Designs for Research*, 1963. Library of Congress Catalogue Card Number: 66:25213
3. Hudson, Tomas. (1997) Walkthrough Project, *University of North Carolina Web Pages,* http://www.cs.unc.edu/~walk/
4. IRIS Performer. (1997) http://www.sgi.com/Technology/Performer/
5. Marrin, C., and Campbell, B. (1997) *Teach Yourself VRML2 in 21 Days*. Sams Net. ISBN 1-57521-193-9
6. Quake. (1997) ID Software http://www.idsoftware.com/
7. Slater, Mel, & Usoh, Martin. Simulating Peripheral Vision in Immersive Virtual Environments, 1993, *Computers & Graphics,* vol 17, No. 6. 643-653.
8. Sperger, Michael. (1996) Virtual Ancient Egypt, *SIMLAB Web Pages,* http://topaz.rec.ri.cmu.edu/files/egypt/.
9. Turvey, M. T. and Kugler, P. N. (1987) *Information, Natural Law, and the Self-assembly of Rhythmic Movement*, Lawrence EarlBaum Associates. ISBN 0-89859-640-8