Technical Report no. 2006-19

On Navigation Guidance for Exploration of 3D Environments

Niklas Elmqvist *Philippas Tsigas*

CHALMERS | GÖTEBORG UNIVERSITY

Department of Computer Science & Engineering
Chalmers University of Technology and Göteborg University
412 96 Göteborg, Sweden

Göteborg, 2006

Abstract

Navigation in complex and large-scale 3D virtual environments has been shown to be a difficult task, imposing a high cognitive load on the user. In this paper, we present a comprehensive method for assisting users in exploring and understanding such 3D worlds, computing first a complete path through the world used as a basis for exploration and then supporting the rich interactions necessary for navigating and wayfinding in the world. The method consists of two distinct phases: an off-line computation step deriving a grand tour using the world geometry and any semantic target information as input, and an on-line interactive navigation step providing guided exploration and improved spatial perception for the user. The former phase is based on a voxelized version of the geometrical dataset that is used to compute a connectivity graph for use in a TSP-like formulation of the problem. The latter phase takes the output tour from the off-line step as an input for guiding 3D navigation through the environment using a technique we call spring-zooming. A user study indicates a significant efficiency improvement in performing visual search tasks in a complex 3D environment using the technique in comparison to unaided 3D navigation. Furthermore, the results show that the spring-zooming technique strikes a good balance between guidance and interaction, achieving significantly better general recall performance in comparison to a simple tour-following technique allowing for no user control.

Keywords: tour generation, navigation aid, navigation assistance

1 Introduction

Spatial understanding of the structure of a 3D virtual world is vital for a user to be able to navigate and solve tasks efficiently, yet this understanding is exceedingly difficult to attain as the worlds become increasingly complex and increasingly transient. New advances in technology allow designers to increase the visual realism (and thus also the visual complexity) of their 3D worlds to hitherto unseen levels, exacerbating this problem. Furthermore, many worlds are today dynamically created for a specific purpose, such as in response to a search query or as the result of a computation, and will exist only for the duration of the interaction. Our users must then be regarded as tourists in these worlds, lacking specific knowledge about the environment they are exploring, yet in need of solving their tasks as rapidly as possible. There is an obvious conflict in this state of being.

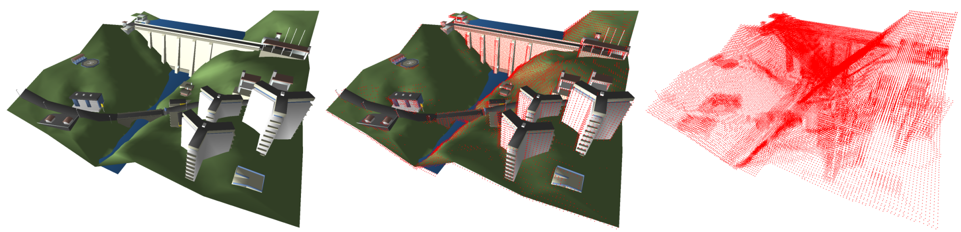

Figure 1: Voxelization process for a complex 3D environment.

In this paper, we propose to bridge this gap between the prevalence of complex and unknown 3D worlds, and the user desire to navigate and traverse these worlds effortlessly, using computer-supported navigation guidance. In essence, instead of forcing the user to expend precious time learning a 3D environment, we devise a method to let the computer first explore the environment and extract the vital paths prior to presenting the environment to the user. We then use this information to augment the standard navigation controls, essentially "holding the user's hand" as he or she traverses the world. Depending on the level of interaction desired by the user, we can impose constraints on the path, speed, deviation, and camera direction as the user moves through the world. Furthermore, even if the user wants to navigate freely, the path information can be used to smoothe the user's ride, avoid jarring collisions (if collision detection is implemented) or disorienting ghosting through walls (if no collision detection), and ensure that the user visits all targets.

The main contributions of this paper are the following: (i) an off-line method for automatically computing a "good" tour through a general 3D environment; (ii) an on-line 3D navigation assistance technique based on path data from the off-line step and providing variable interaction; and (iii) an empirical user study evaluating the effectiveness of the new on-line navigation technique in comparison to standard unaided 3D navigation.

The application domains of this method are many and varied: it can be used for visual storytelling when introducing a new 3D environment, familiarizing a 3D modeller or designer with an unknown or half-forgotten project, presenting all the relevant information in a visualization space, and more. The off-line computation step is designed to be as efficient as possible, providing acceptable tour information with a minimum of time investment. The on-line component can be configured to either be unobtrusive, merely nudging the user in the right direction, or take full control of the user's movement through the world.

This paper is structured as follows: We begin with a review of existing work on 3D navigation in general and navigation assistance in particular. The following two sections describe the off-line tour generation and the on-line 3D navigation assistance techniques, respectively. We describe our user study and the results, and finish the paper with a discussion and conclusions of our findings.

2 Related Work

Effective navigation through a three-dimensional computer environment is a well-known problem that has been attacked from many different directions over the last few years. The problem arises for any environment larger than what can be seen from a single viewpoint, forcing the user to rely on a mental representation of spatial knowledge, often called a *cognitive map* [5, 23]. Generally speaking, while navigation can be a challenging task even in the physical world, the absence of many sensorial stimuli in the virtual world compounds the problem even further [6].

Wayfinding is typically defined as a cognitive aspect of navigation with the purpose of planning and forming strategies prior to executing them, i.e. where the actual navigation is not the goal of the interaction but the means to solve some specific task [10]. The wayfinding task is conducted on the user's cognitive map, and thus it is clear that if the user lacks an accurate mental representation of the environment, performance will suffer.

2.1 Spatial Design

One approach to improve wayfinding is to organize the virtual environment in a way that promotes understanding and orientation, in essence making it easier for the user to construct an accurate cognitive map. Due to the similarities with navigation in physical space [24], we can leverage existing research from urban planning, geography, and psychology. For example, Darken and Sibert suggest a number of design guidelines for organizing a virtual environment to facilitate the acquisition of spatial knowledge [10], further extended in [9]. Similarly, Vinson [24] argue for the importance of landmarks for navigation in a 3D world, and give a comprehensive set of guidelines for their placement, design, and composition.

2.2 Navigation Widgets

Visual aids can be used to great effect for improving 3D navigation. Chittaro and Burigat [6] present an array of different compass-like navigation widgets for helping the user to find important objects and places in a virtual environment. Trails [20] help users utilize previous explorations to improve their current search. Path drawing [15] lets the user draw an intended path directly on the 2D view of the world to aid navigation.

2.3 Motion Control

Another powerful class of navigational aids is motion control, i.e. different methods of traveling through a virtual environment and potentially guiding or constraining the user's movement. Techniques in this class can have varying degrees of obtrusiveness, from merely nudging the user in the right direction to constraining or downright controlling the viewpoint completely. Bowman et al. [4] present a taxonomy of first-person motion control techniques for manual viewpoint travel that is useful for evaluating such methods.

The flying, eyeball-in-hand, and scene-in-hand metaphors [25, 26] constitute perhaps the most basic motion control techniques with practically no automatic control. Mackinlay et al. [18] describe a method of logarithmically controlling the viewpoint speed while moving through a 3D world to allow for rapid motion over large distances, yet slowing down when approaching the target. In related work, Song and Norman [21] propose a set of non-linear motion control techniques for intuitively traversing virtual environments. The work of Tan et al. [22] on a moded navigation technique is interesting, not only for the fact that it contextually combines two different motion control techniques (flying and orbiting), but also that it couples the speed of movement to the height and tilt of the camera to smoothly support both local detail views and global overviews.

Guided navigation techniques exhibit a little more control on the motion of the viewpoint, allowing the computer to augment the user's spatial knowledge with additional information. Wernert and Hanson [27] present a taxonomy of assisted navigation, and also discuss a "dog-on-a-leash" approach to guidance through a 3D world. This approach is similar to the "river analogy" introduced by Galyean [13], where the viewpoint is tethered to a vehicle following a path through the virtual environment and some degree of control is retained by the user. The motion control technique presented in this paper is similar to both the river and dog metaphors, yet supports variable interaction to a higher degree. Furthermore, unlike these two papers, our work also includes an empirical user study comparing both the effectiveness of guided navigation over unguided navigation, as well as the impact of user control on world recall.

Another notable technique is the virtual guide of Chittaro et al. [7] which the user must follow actively; the guide's path is also automatically computed using an algorithm operating on a 2D occupancy matrix similar to the tour generation algorithm in this paper, but our method can handle any general 3D environment and not just one-floor buildings.

Finally, constrained navigation techniques essentially assume full control of viewpoint motion, sometimes even moving the gaze of the user in the desired direction. By reducing the freedom of the user, navigation and wayfinding can be simplified, and reduce the need for expensive features such as collision detection. Examples of this approach include that of Hanson and Wernert [14], who employ invisible surfaces to constrain user movement, and of Andújar et al. [2], whose Way-finder system algorithmically computes an exploration path through a 3D environment. The latter algorithm is based on a voxelized version of the 3D world, just like the tour generation algorithm presented in this paper, but uses another method to compute a cell and portal graph for use with a backtracking tour generator, whereas our algorithm builds disjoint visibility subsets and performs TSP computations on the resulting connectivity graph.

3 Overview

Historically, the term "grand tour" used to refer to the peculiar rite of passage that young European (typically British) noblemen undertook more or less as part of their education during the late 1600s until the 1800s. The tour was essentially a travel itinerary of Europe, designed to expose the neophyte to as many of the important cultural and historical landmarks as possible [12]. In scientific visualization, on the other hand, a grand tour is a method for viewing multidimensional data using orthogonal projections onto a sequence of lowerdimensional subspaces [3].

In the context of this paper, the term is perhaps more literally related to the historical use of the term than the mathematical method. The basic idea is to take the user on a sightseeing tour of a certain 3D world in order to help him or her in understanding its structure and important landmarks. For this purpose, the tour has been designed so that it visits all of the landmarks in some suitable order. It can either be built manually by a human designer, or computed automatically using a tour generation algorithm (the approach taken in this paper). The tour is then used for guiding the user in interactively exploring the world.

See Figure 2 for an overview of the two-step process presented in this paper. The following sections will give the details on these two phases.

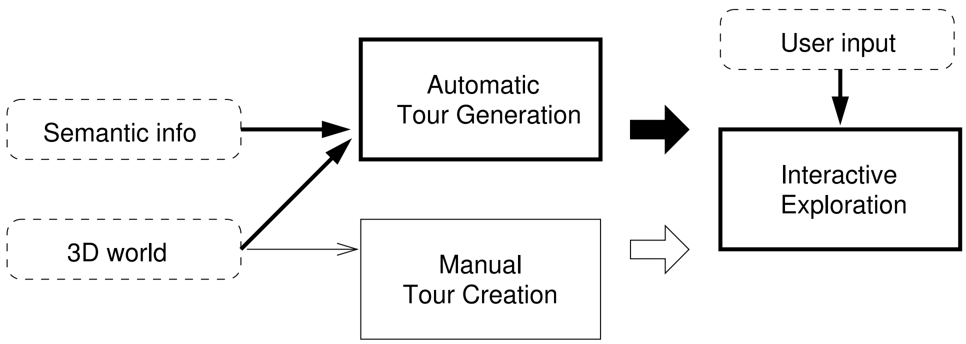

Figure 2: Navigation guidance overview.

4 Automatic Tour Generation

The objective of the automatic tour generation phase is to build a grand tour of a 3D world given the following input:

- a 3D world geometry dataset;
- a set of landmarks; and
- a starting point.

The tour should start and end in the starting point and visit all of the landmarks in the world. Beyond these simple requirements, we can add a few more: the generated tour should be "good" in some sense, and the process should be robust in the presence of inaccessible landmarks, i.e. landmarks that are landlocked and cannot be visited due to surrounding geometry.

The definition of a "good" tour is open to debate; in this work, we take it to mean a tour of as short length as possible (not necessarily optimal) that visits all landmarks as few times as possible. Furthermore, the tour should not stray outside the bounding box of the 3D world to avoid trivial (but impractical) solutions where the viewpoint is placed at infinity.

An important observation is that visiting a landmark in this context is equivalent to seeing it, so it is not necessary (and in fact undesirable) to pass through the same spatial location as the landmark for it to be regarded as having been visited. At the same time, our algorithm allows for specifying a maximum visibility distance, i.e. the furthest away the tour may pass a landmark in order to visit it.

Finally, the representation of landmarks is significant; in our implementation, we choose to use 3D points for simplicity, but the algorithm can support other 3D primitives as well as actual 3D triangle meshes as targets.

Figure 4 gives a rough outline of the sequential tour generation process. We explain each of these steps in more detail in the following subsections.

4.1 Voxelization

Our tour generation algorithm operates on a voxelized version of the 3D world, so the initial step of the process is to voxelize the geometry dataset into a volume representation (see Figure 1 for an example). We first compute the bounding box of the world and enlarge it in all directions by a single voxel width to allow for the algorithm to skirt along the perimeter of the 3D world if necessary. Then we voxelize the world using incremental 3D scan-conversion.

The process of incremental 3D scan-conversion builds a volume representation of a 3D boundary representation such as a triangle mesh by iteratively scan-converting the 3D primitives into a voxel buffer. Kaufman and Shimony [16] give algorithms for scan-converting all manners of 3D primitives; our method is based on a recursive subdivision of 3D space into an octree representation and testing the triangle against each volume using a fast triangle-box intersection test [1] (see Figure 3).

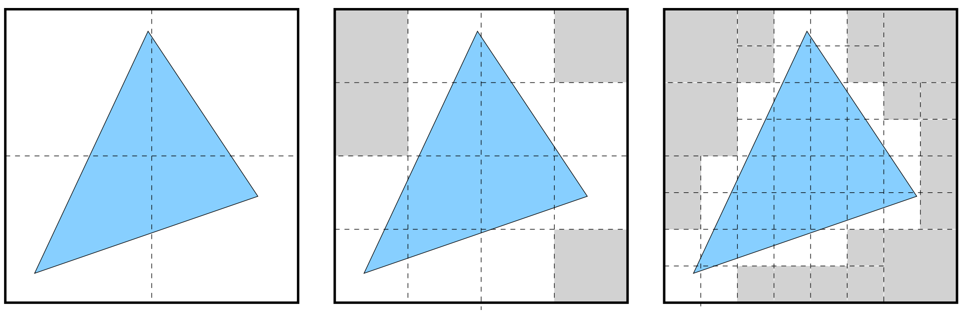

Figure 3: Recursive 3D scan-conversion using an octree.

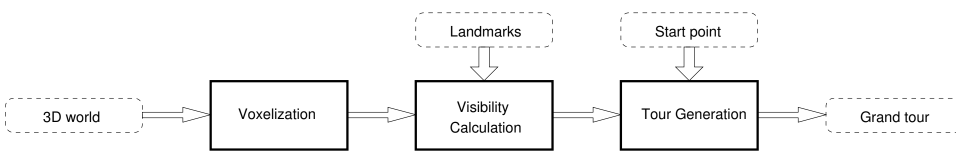

Figure 4: The tour generation process.

4.2 Visibility Calculation

Armed with a volume representation of the 3D dataset, we can now calculate the visibility information of all voxels given the set of landmarks the tour should visit. We use an integer one-pass voxel traversal algorithm [17] to determine if there is a clear line of sight between the current voxel and a specific landmark (this particular step must be generalized for landmarks represented as something more complex than a point). Additional constraints can also be imposed at this point; we currently ensure that the distance between the voxel and the landmark is within the maximum visibility distance, but other constraints are plausible.

Having derived the visibility information for all voxels, we then group them into disjoint subsets we call *visibility sets* using a breadth-first search algorithm. Each set is built so that its members are contiguous and have the same visible landmarks. A special case is made for voxels with no visible landmarks; they form "zero-visibility" sets, and are necessary for connectivity in the world.

The visibility sets together form a connectivity graph specifying the general visibility structure of the 3D world. At this point, it is possible to subject the connectivity graph to an optional optimization step. Many visibility sets are redundant or useless and may be removed from the graph; examples include zero visibility leaf nodes as well as nodes whose visibility is subsumed by its neighbors. Care must be taken not to remove nodes so that the graph no longer is connected, however.

The final step of the visibility calculation phase is to identify the *border voxels* for each visibility set, i.e. the voxels that are adjacent to voxels in another set. We know that that in order to travel from one neighbor to another through a specific node, the tour will have to pass at least one voxel in each border set. Again we can optimize the problem (but this time by an approximation); our implementation identifies a single "entry point" border voxel for each neighbor by minimizing its average distance to the other neighbors in the visibility set together with its counterpart border voxel in the neighboring set.

4.3 Tour Generation

The stage is now set for generating the tour through the 3D environment. We use a TSP-like formulation of the problem. It is important to remember that it is not necessary to visit all of the visibility sets in the connectivity graph (as in traditional TSP), just enough to cover the all of the landmarks. Thus, we can model our problem as what is known in the literature as a *Generalized Travelling Salesman Problem* (GTSP), where the n nodes in the undirected graph G are partitioned into m disjoint subsets called *clusters*, and where it is sufficient to visit only one node in each cluster.

Given that GTSP reduces to TSP when $m = n$, GTSP is clearly NP-hard, so in our implementation we do not aim for an optimal solution of the problem. Instead, we use the border voxels computed in the previous phase to reformulate the connectivity graph as a *border graph* with the border voxels as nodes and the interior of the visibility sets as edges. The length of the edges connecting border voxels can either be found using a shortest-path algorithm such as A^* or simply approximated by the Euclidean distance. See Figure 5 for an example of a simple border graph for 3D world represented by four visibility sets (A, B, C, and D) and with the paired border voxels as white boxes.

Using this representation, we can now employ a standard heuristic [8] based on computing the minimum spanning tree of the connectivity graph and deriving a Hamiltonian cycle from it. Our unique modification is the added termination condition to quit when all landmarks have been visited. In Figure 5, with a starting point in visibility set A, it is easy to see that a tour would proceed in the following order: A, C, B, C, D, C, A.

Finally, the last step of our tour generation phase is to derive a detailed voxel-level tour given the connectivity graph tour computed in the previous step. We do this by iteratively moving along the graph tour, calculating the shortest path from the current position to any border voxel in the next visibility set to visit. Each such instance is constrained to the particular visibility set, cutting down the search space considerably.

4.4 Performance

Performance measurements of the tour generation phase applied to the four different scenarios from Section 6.5 are presented in Table 1. The measurements were conducted on a dual-processor Intel Xeon 3 GHz computer with 1 GB of RAM. The main bottleneck of the algorithm is the last step, i.e. the derivation of local paths within the voxel sets. Currently, this is performed using a variant of Dijkstra's

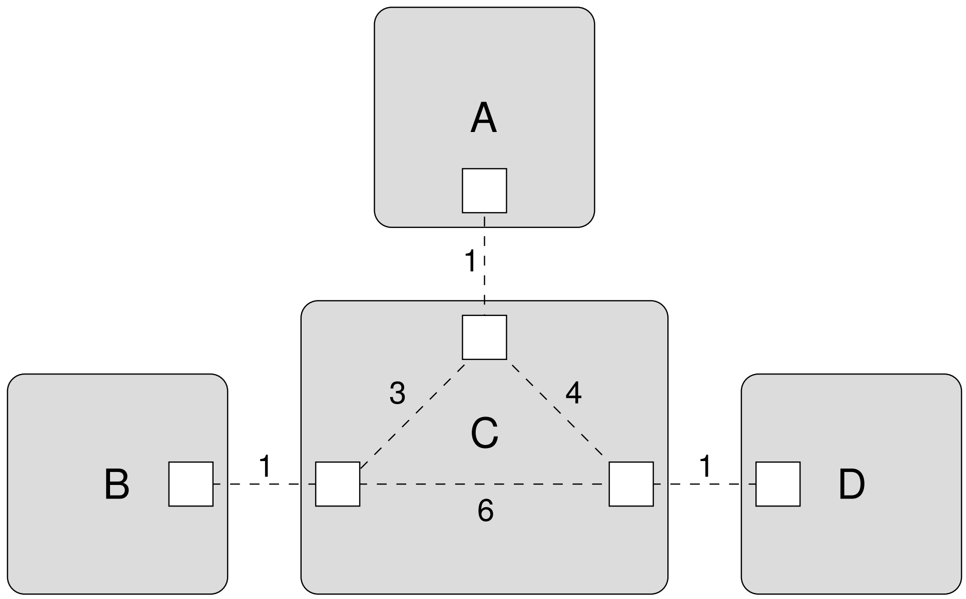

Figure 5: Border graph representation of a 3D world.

shortest path algorithm, but more complex and optimized solutions are certainly possible.

As can be seen from the results, the visual complexity of the scene is more or less irrelevant; the voxelization phase is a very small fraction of the total time. Rather, the important metric is the degree of occlusion in the world. For the indoor scenario, the occlusion is high despite high visual complexity, resulting in fast computation. For the outdoor scenario, on the other hand, its open nature yields very large visibility sets, causing high computation time.

The voxel size can be used to somewhat control both computation time as well as memory consumption; the larger the voxels, the shorter computation time and the less memory is used. On the other hand, larger voxel size implies a less accurate volume representation, causing the quality of the generated tours to suffer.

Scenario	**Triangles**	**Time**
outdoor	558,130	9 minutes 4 seconds
indoor	484,673	59 seconds
infoscape	12,844	7 minutes 49 seconds
conetree	16,576	2 minutes 1 second

Table 1: Tour generation performance for the four scenarios.

5 Guided 3D Navigation

Our method for 3D navigation guidance is designed to both help the user in discovering all of the specific landmarks in the world, as well as helping her in the building of an accurate cognitive map of the world as a whole. To achieve the former, we employ a grand tour of the world, either created manually by a human designer or generated automatically by an algorithm such as the one described above. To achieve the latter, we allow the user to retain some control over her movement along the tour, seeking to engage the user as an active participant in the exploration.

We call our guided exploration technique *spring-zooming*, inspired by the spring-like umbilical cord that the viewpoint is connected to the grand tour with. See Figure 6 for a schematical overview. Depending on the level of interaction desired, we can impose variable constraints on the following properties:

- **Speed.** Movement along the tour can either be computer-controlled or user-controlled—in our implementation, the up and down arrow keys are used to start and stop movement forwards or backwards along the tour.

- **Viewpoint direction.** The direction of the camera can either be slaved to the direction of movement, fixed to follow the currently closest landmark, or fully user-controlled (hybrids are possible).

- **Local deviation.** To facilitate active participation, we can allow deviations from the tour path using the spring-zooming technique. Using a simple interaction technique, the user can smoothly zoom the viewpoint forward or backwards in the direction of movement to the full extent of the connecting spring (using the center and right mouse buttons in our implementation).

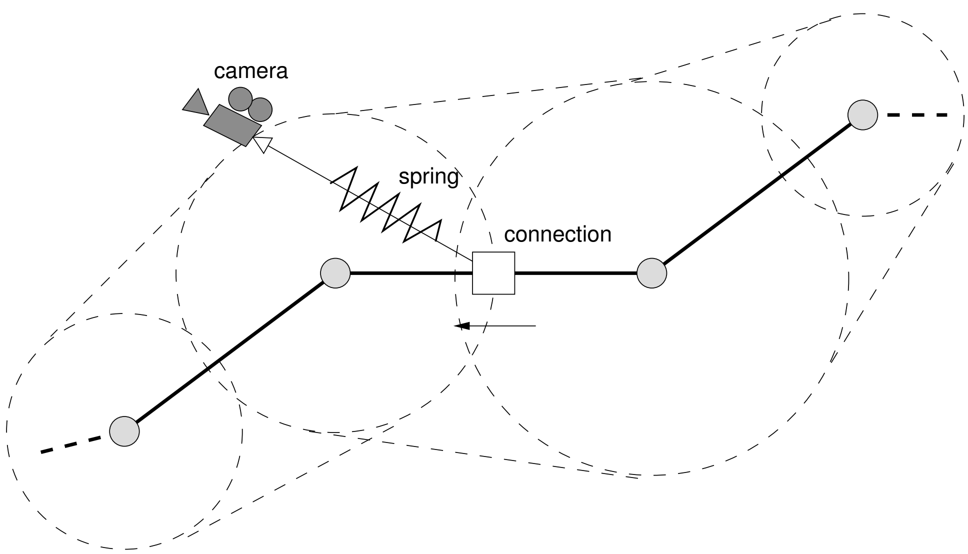

Figure 6: Spring-zooming overview (the circles show the free space around each node.)

Depending on whether collision detection is enabled or not, the viewpoint may either collide when it comes into conflict with world geometry, or it may float through the geometry as if it was not there. These two events, called collisions and ghosting, may be potentially disorienting to the user, and is typically a major complaint when exploring a 3D world. Avoiding these occurrences is a secondary objective of the spring-zooming technique, and it is done by computing the amount of free space around the tour in all points and constraining the full length of the umbilical cord to this value. This ensures a smooth and continuous ride through the environment with no jarring stops or confusing ghosting.

The grand tours accepted as input are generally discrete waypoints in space, and so we fit Hermite curves [11] to these points to smoothe the movement through the 3D space.

6 User Study

The basic premise of this research is that guiding the user in exploring a 3D world will increase the user's efficiency in solving visual search tasks compared to unguided navigation. However, we also hypothesize that fully constraining the movement of the viewpoint will reduce the viewer to a passive recipient instead of an active participant, somewhat akin to being a passenger in a car as opposed to driving the car yourself. Accordingly, the user's perception of the world as a whole will suffer even if he or she is shown the important landmarks by the guidance technique.

To evaluate these two statements, we conducted a formal user study exposing a number of subjects to visual search tasks in four different types of environments with a recall phase designed to test general familiarity with the environment and an evaluation phase for measuring visual search performance.

6.1 Subjects

We recruited 16 subjects for this study, four of which were female. The subjects were all undergraduate and graduate students from the engineering programs at our university. Ages ranged from 20 to 50 years. All participants were screened to have at least basic computer skills, were not color blind, and had normal or corrected-to-normal vision. 9 out of 16 subjects had extensive 3D experience.

6.2 Equipment

The experiment was conducted on an Intel Centrino Duo laptop computer equipped with 2048 MB of memory running the Microsoft Windows XP operating system. The display was a 17-inch widescreen LCD display running at 1920×1200 resolution and powered by an NVIDIA Geforce 7800 GO graphics card.

6.3 Procedure

Each test session lasted approximately one hour, and started with an introduction and a training session to allow the subject to get familiarized with the test application, the interface, and the test procedure. After the subjects indicated that they were satisfied, we proceeded with the actual scenarios. In order to increase the generality of the study, we included four different scenarios designed to mimic various contexts where navigation guidance may be used.

Each scenario was tested in a three-phase sequence: familiarization, recall, and evaluation. In the first phase, users were given the scenario world and were allowed to familiarize themselves with it for five minutes. During this phase, the actual guidance method selected for the user was active. The subject was given a reference card with pictures of three types of landmarks relevant to the actual scenario that he or she should be looking for. An overhead map of the world with the user's own position and location marked was available in the upper left corner of the display.

After five minutes, the experimenter moved on to the recall phase, where the subject was shown a large overhead map of the world and was asked to place as many instances of two of the three target landmarks they could remember. There was no time limit here.

Finally, in the third phase, the subject returned to the 3D world with the task to collect as many as possible of the third type of landmark. Here all subjects were forced to navigate freely with no guidance support. Collecting an object was done by approaching to within a distance of 5% of the world scale and pressing the Tab key. This removed the object from the landscape. The miniature overhead map

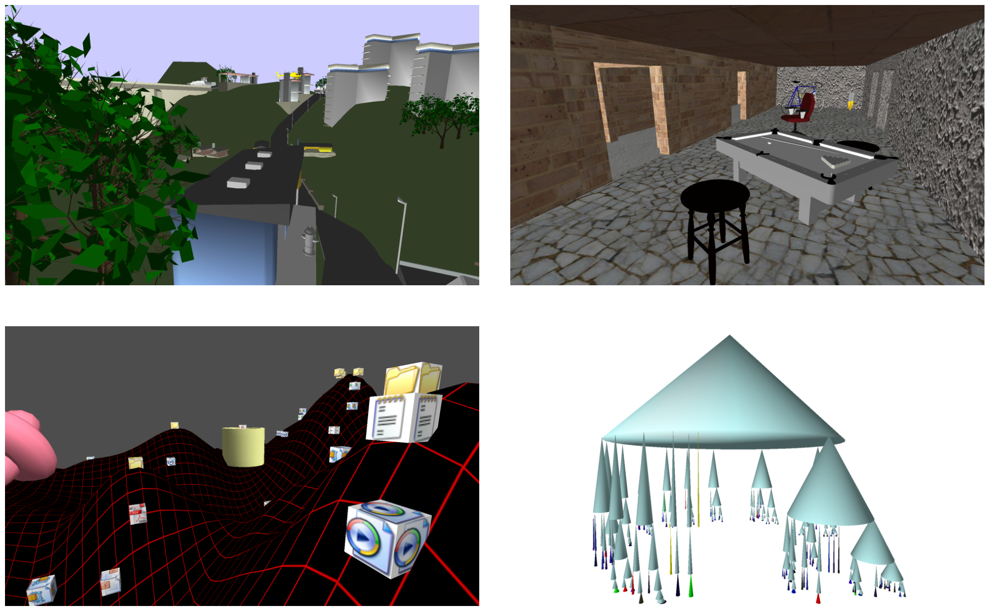

Figure 7: The four scenarios employed in the user study (from left to right: outdoor, indoor, infoscape, and conetree).

was available in this phase as well. When the subject decided that all targets had been found, he or she was able to end the scenario (stopping the time).

Subjects received the four scenarios in counterbalanced order to manage systematic effects of practice. The subjects did not know in advance which two of the target types they would be asked to place in the second phase, nor which landmark to collect in the third.

6.4 Navigation Methods

The navigation method employed was one of the following three:

Free. Unaided first-person 3D navigation with no guidance. The mouse panned the view and the arrow keys moved in the direction of viewing (left and right for strafing).

Follow. Passive tour following with full guidance except for camera orientation. The mouse panned the view.

Spring. Full spring-zooming with user-controlled movement, deviation, and camera orientation. The mouse panned the view, the center and right mouse buttons engaged forward and backward zooming, and the up and down arrow keys controlled movement along the tour.

The free navigation method was used for all subjects in the third phase (evaluation).

6.5 Scenarios

The four different scenarios employed in the experiment were designed to depict typical usage situations of 3D worlds and 3D navigation using both abstract as well as realistic environments. Subjects were given a concrete explanation of the scenario prior to starting each scenario run. Below follows a short description of each scenario (see the example screenshots in Figure 7, ordered left to right):

Outdoor. Large-scale outdoor world with a rescue mission scenario where the user was asked to identify helicopters, cars, and fire hydrants. [realistic]

Indoor. Maze-like single-floor indoor environment representing a furniture store where the user was looking for office chairs, sofas, and floor lamps. [realistic]

Infoscape. Abstract information landscape for a hypothetical 3D file browser where the subject was preparing for writing report by looking for Word, PDF, and Excel files. [abstract]

Conetree. Abstract conetree [19] visualization for the organization hierarchy of a company where the subject was asked to look for leaf nodes of specific colors. [abstract]

6.6 Design

The experiment was designed as a between-subjects comparative study on the independent variable Method (the navigation method used), and with a within-subjects independent variable Scenario (the type of environment). The method variable used the three levels from Section 6.4, i.e. "free", "follow", and "spring". The dependent variables were the relative error (i.e. the number of missed landmarks divided by the total number of landmarks) and average error distance for the recall phase, and the relative error and average time per found landmark for the evaluation phase.

7 Results

In general, the results from the user study confirmed our belief that subjects with navigation guidance would be more efficient at solving the visual search task than those without guidance; the time per found landmark was 43.8 (s.d. 30.4) seconds for free navigation, 19.5 (s.d. 10.7) seconds for follow navigation, and 17.6 (s.d. 11.9) seconds for spring-zooming (see Figure 8). This difference was also significant: $t(42) = 3.398, p = .001$ for the comparison between free and follow, and $t(42) = 3.624, p = .001$ for free versus spring. The difference between follow and spring was not significant ($t(38) = .529, p = .600$).

Furthermore, subjects using spring-zooming were also more correct in collecting landmarks in the evaluation phase than those using free navigation: the error rate was .198 (s.d. .245) for free navigation, .123 (s.d. .160) for follow mode, and .058 (s.d. .130) for spring-zooming. The difference was only significant for free versus spring ($t(42) = 2.297, p = .027$) and not for free versus follow ($t(42) = 1.179, p = .245$) nor for follow versus spring ($t(38) = 1.400, p = .170$).

What was more surprising was that subjects were more accurate in placing landmarks in the recall phase for spring-zooming than for the other two methods; the average error distance per landmark (normalized using the scale of the world) was .236 (s.d. .124) for free navigation, .192 (s.d. .102) for passive follow navigation, and .099 (s.d. .073) for spring-zooming. The difference was significant between free and spring ($t(42) = 4.335, p < .001$) as well as follow and spring ($t(42) = 2.297, p = .027$), but not between free and follow ($t(38) = 3.282, p = .002$). This goes against our hypothesis that the navigation methods which permit some measure of user control (i.e. free navigation and spring-zooming) would promote significantly better recall than passive tour following; as it turned out, spring-zooming was significantly more accurate than both other methods.

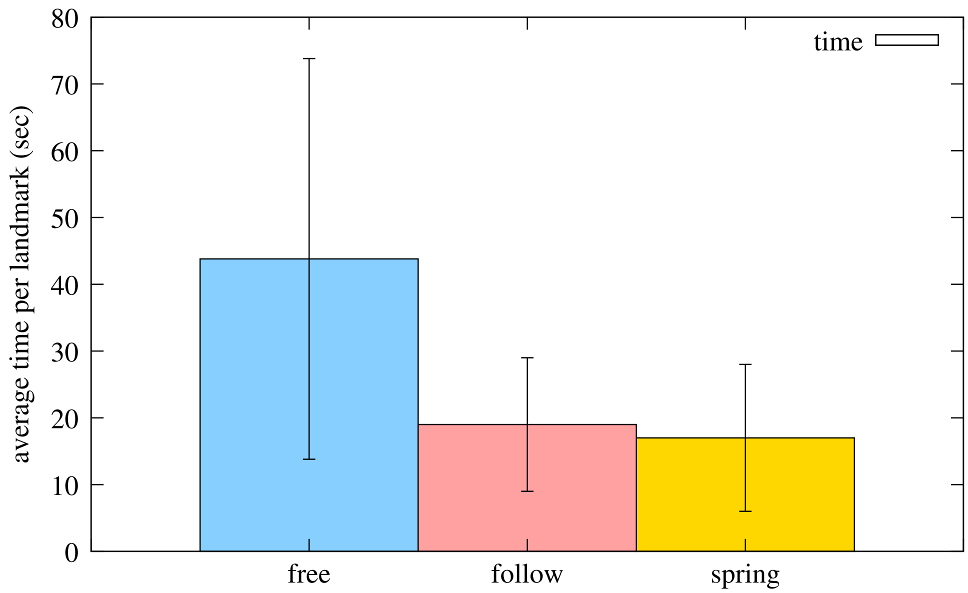

Figure 8: Performance results for the evaluation phase (average time per found landmark). Error bars show standard deviations.

Finally, as for correctness in the recall phase, no conclusive results were found; the relative error rate was .238 (s.d. .191) for free navigation, .174 (s.d. .173) for follow mode, and .191 (s.d. .201) for spring-zooming (see Figure 9). None of these differences were significant; $t(42) = 1.146, p < .258$ for free versus follow, $t(42) = .782, p = .439$ for free versus spring-zooming, and $t(38) = -.289, p = .774$ for follow versus spring-zooming.

8 Discussion

While the results from the user study confirmed our basic hypothesis that navigation guidance will improve search performance over free navigation, it was a little bit surprising that our second hypothesis on user control promoting the formation of a cognitive map was not confirmed. One possible explanation might be that the subjects in the passive follow group were not in fact passive recipients since they were given a very specific task when familiarizing themselves with the 3D world. Therefore, they performed better than they might have done without this knowledge. However, our pilot testing showed that the alternative, i.e. not telling the subjects which kinds of landmarks to look for, was simply not feasible for the high-detail scenarios we used in the study.

Regrettably, two of the participants in the user study became motion sick (one still finished the study, the other was forced to cancel). An interesting observation is that both of these participants were assigned to the passive tour following group—a plausible (if perhaps unfounded) explanation may be that users that have no control over their movement run a greater risk of this, somewhat akin to how people who are prone to motion sickness while riding cars typically only get it when they are passengers and not driving themselves.

Due to space concerns, only a summary of the results were presented in this paper. We have also analyzed the results based on the scenario, and these indicate the same general trends as the overall results. It is worth noting that there was no

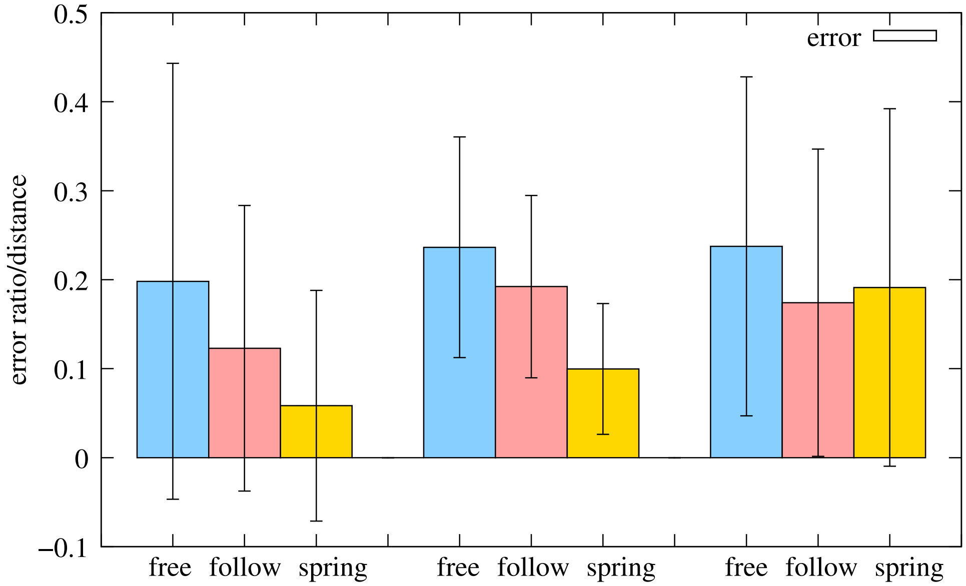

Figure 9: Correctness results (evaluation error rate, recall error distance, recall error rate). Error bars show standard deviations.

significant differences between any of the dependent variables in the indoor scenario, an environment with a high degree of occlusion. For the conetree scenario, on the other hand, an environment with a low degree of occlusion, the average time per target was not significant for any method (recall distance still was for spring-zooming, however).

9 Conclusions

We have presented a method for navigation guidance in the exploration of general 3D environments intended to both promote the user's building of a cognitive map of the environment as well as to improve visual search task performance. The method works for both abstract as well as realistic visualizations and operates in two distinct steps: an off-line tour generation step that builds a grand tour of the given 3D world that visits all of its landmarks, and an on-line interactive navigation technique that guides the user along the tour while still allowing for some user control. This last step is vital in order to make the user an active participant in the navigation. A user study was conducted to investigate the impact of the new technique compared to free navigation as well as passive tour following, and the results indicate a significant improvement for both search performance and general recall for the new technique.

Acknowledgments

Thanks to Robert Karlsson for supplying some of the 3D models that were used in the user study scenarios. Thanks to the anonymous test subjects who participated in the study.

References

[1] Tomas Akenine-Möller. Fast 3D triangle-box overlap testing. *Journal of Graphics Tools*, 6(1):29–33, 2001.

[2] Carlos Andújar, Pere-Pau Vázquez, and Marta Fairén. Way-finder: guided tours through complex walkthrough models. In *Proceedings of EUROGRAPHICS 2004*, pages 499–508, 2004.

[3] D. Asimov. The grand tour: A tool for viewing multidimensional data. *SIAM Journal on Scientific and Statistical Computing*, 6(1):128–143, January 1985.

[4] Doug A. Bowman, David Koller, and Larry F. Hodges. Travel in immersive virtual environments: An evaluation of viewpoint motion control techniques. *Proceedings of the IEEE Conference on Virtual Reality 1997*, pages 45–52, 1997.

[5] W. G. Chase. Visual information processing. In K. R. Boff, L. Kaufman, and J. P. Thomas, editors, *Handbook of Perception and Human Performance, Vol II: Cognitive Processes and Performance*, pages 28–1–28–71. John Wiley and Sons, 1986.

[6] Luca Chittaro and Stefano Burigat. 3D location-pointing as a navigation aid in virtual environments. In *Proceedings of ACM Conference on Advanced Visual Interfaces 2004*, pages 267–274, 2004.

[7] Luca Chittaro, Roberto Ranon, and Lucio Ieronutti. Guiding visitors of Web3D worlds through automatically generated tours. In *Proceedings of ACM Conference on 3D Web Technology 2003*, pages 27–38, 2003.

[8] Thomas H. Cormen, Charles E. Leiserson, and Ronald L. Rivest. *Introduction to Algorithms*. MIT Press, Cambridge, MA, 1990.

[9] Rudolph P. Darken and Barry Peterson. Spatial orientation, wayfinding, and representation. In Kay M. Stanney, editor, *Handbook of Virtual Environment Technology*. Lawrence Erlbaum Associates, 2001.

[10] Rudolph P. Darken and John L. Sibert. Wayfinding strategies and behaviors in large virtual worlds. In *Proceedings of ACM CHI'96 Conference on Human Factors in Computing Systems*, pages 142–149, 1996.

[11] James D. Foley, Andries van Dam, Steven K. Feiner, and John F. Hughes. *Computer Graphics: Principles and Practice*. Addison-Wesley, Reading MA, 1990.

[12] Paul Fussell. *The Norton Book of Travel*, chapter The Eighteenth Century and the Grand Tour. W.W. Norton and Co., 1987.

[13] Tinsley A. Galyean. Guided navigation of virtual environments. In *Proceedings of the ACM Symposium on Interactive 3D Graphics 1995*, pages 103–104, 1995.

[14] Andrew J. Hanson and Eric A. Wernert. Constrained 3D navigation with 2D controllers. In *Proceedings of the IEEE Conference on Visualization 1997*, pages 176–182, 1997.

[15] Takeo Igarashi, Rieko Kadobayashi, Kenji Mase, and Hidehiko Tanaka. Path drawing for 3D walkthrough. In *Proceedings of the ACM Symposium on User Interface Software and Technology 1998*, pages 173–174, 1998.

[16] Arie Kaufman and Eyal Shimony. 3D scan-conversion algorithms for voxel-based graphics. In *Proceedings of the ACM Workshop on Interactive 3D Graphics 1986*, pages 45–75, 1986.

[17] Y. K. Liu, B. Zalik, and H. Yang. An integer one-pass algorithm for voxel traversal. *Computer Graphics Forum*, 23(2):167–172, 2004.

[18] Jock D. Mackinlay, Stuart K. Card, and Robertson Robertson. Rapid controlled movement through a virtual 3D workspace. In *Computer Graphics (SIGGRAPH '90 Proceedings)*, volume 24, pages 171–176, August 1990.

[19] George G. Robertson, Jock D. Mackinlay, and Stuart K. Card. Cone trees: Animated 3D visualizations of hierarchical information. In *Proceedings of the ACM CHI'91 Conference on Human Factors in Computing Systems*, pages 189–194, 1991.

[20] Roy A. Ruddle. The effect of trails on first-time and subsequent navigation in a virtual environment. In *Proceedings of the IEEE Conference on Virtual Reality 2005*, pages 115–122, 2005.

[21] Deyang Song and Michael Norman. Nonlinear interactive motion control techniques for virtual space navigation. In *Proceedings of IEEE Virtual Reality Annual International Symposium 1993*, pages 111–117, 1993.

[22] Desney S. Tan, George G. Robertson, and Czerwinski Czerwinski. Exploring 3D navigation: Combining speed-coupled flying with orbiting. In *Proceedings of ACM CHI 2001 Conference on Human Factors in Computing Systems*, pages 418–425, 2001.

[23] Edward C. Tolman. Cognitive maps in rats and men. *The Psychological Review*, 55(4):189–208, July 1948.

[24] Norman G. Vinson. Design guidelines for landmarks to support navigation in virtual environments. In *Proceedings of the ACM CHI'99 Conference on Human factors in Computing Systems*, pages 278–285, 1999.

[25] Colin Ware and Danny R. Jessome. Using the bat: A six-dimensional mouse for object placement. *IEEE Computer Graphics and Applications*, 8(6):65–70, November 1988.

[26] Colin Ware and Steven Osborne. Exploration and virtual camera control in virtual three dimensional environments. *Computer Graphics*, 24(2):175–183, 1990.

[27] Eric A. Wernert and Andrew J. Hanson. A framework for assisted exploration with collaboration. In *Proceedings of the IEEE Conference on Visualization 1999*, pages 241–248, 1999.