Interactive Exploration Assistance for Immersive Virtual Environments Based on Object Visibility and Viewpoint Quality

Sebastian Freitag* Benjamin Weyers* Torsten W. Kuhlen*

Visual Computing Institute, RWTH Aachen University, Germany — JARA-HPC, Aachen, Germany

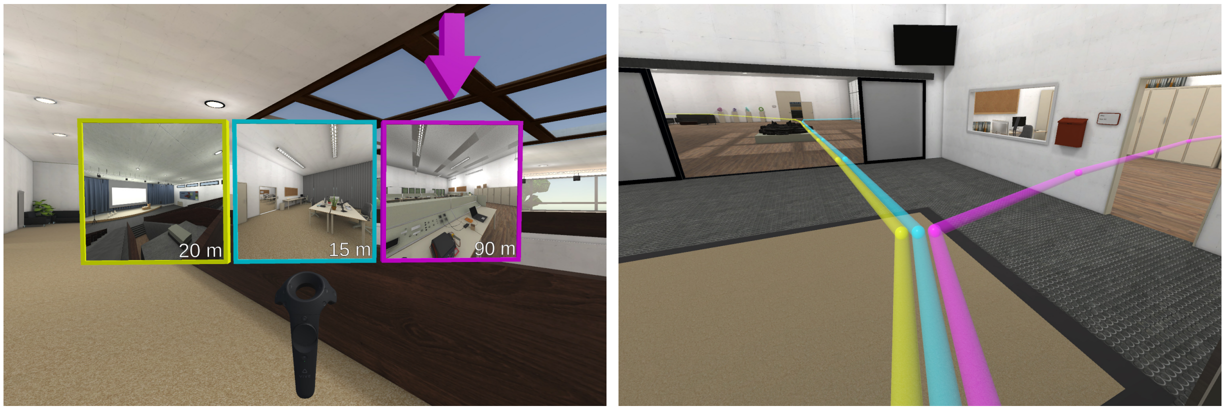


Figure 1: Left: The interface shows "photos" of three suggested target locations above the user's controller. Arrows are displayed above each if the target is significantly above or below the user. Right: The suggested path to each target is visualized as a color-coded tube.

Abstract

During free exploration of an unknown virtual scene, users often miss important parts, leading to incorrect or incomplete environment knowledge and a potential negative impact on performance in later tasks. This is addressed by wayfinding aids such as compasses, maps, or trails, and automated exploration schemes such as guided tours. However, these approaches either do not actually ensure exploration success or take away control from the user.

Therefore, we present an interactive assistance interface to support exploration that guides users to interesting and unvisited parts of the scene upon request, supplementing their own, free exploration. It is based on an automated analysis of object visibility and viewpoint quality and is therefore applicable to a wide range of scenes without human supervision or manual input. In a user study, we found that the approach improves users' knowledge of the environment, leads to a more complete exploration of the scene, and is also subjectively helpful and easy to use.

Index Terms: Human-centered computing—Human computer interaction—Interaction paradigms—Virtual reality; Computing methodologies—Computer graphics—Rendering—Visibility

1 Introduction

One of the first tasks carried out by a user when entering an unknown virtual scene is often *exploration*, the act of acquiring knowledge of the environment, which is an important prerequisite for many subsequent tasks depending on this knowledge [27]. However, during free exploration, important parts of the scene are often overlooked or forgotten, leading to incorrect or incomplete environment knowledge and a potential negative impact on user performance in subsequent tasks.

Therefore, various approaches to facilitate exploration have been proposed. For example, an appropriate scene design supporting wayfinding can contribute to exploration success, e.g., by featuring high environment legibility [28], landmarks [32, 43], or signs [27]. However, since modifying the virtual world is usually only possible for the scene designer and restricted to certain circumstances, its use is limited. Instead, for arbitrary scenes, exploration can be supported by including wayfinding aids in user interfaces, e.g., using tools such as compasses [5] or maps [8, 9, 44], or by showing the user's previous path [9, 20, 35]. However, while these aids are in many cases applicable to a wide range of scenes, they only provide the user with supportive tools, but do not ensure the actual exploration success, e.g., by preventing that important areas are missed.

An approach to ensure successful and efficient global scene exploration are automated virtual tours, for which a camera path through the scene is computed that visits each important location at least once. This can be prepared by either defining these locations manually (e.g., [12]), by determining them using viewpoint quality (e.g., [1, 41])—an automatic metric for the informativeness of each point in the scene [36, 38, 48]—or by combining both (e.g., [40]). In addition to static tours, it has also been proposed to introduce some interactivity, e.g., by allowing the user to deviate to some degree from the prescribed path (e.g., [13, 19]). However, virtual tours usually force the user to follow either the complete predetermined path or some part of it, and even if they are (partially) interactive, do not consider where the user has already explored on their own. Therefore, they are not well suited to only supplement free exploration by making sure that the user does not miss anything interesting or important.

Therefore, in this work, we present an approach for an interactive exploration assistance interface that can be used as an extension to regular, free exploration using any ground-based travel technique. Based on an analysis of which parts of the scene the user has already seen and which places are likely to be interesting according to viewpoint quality, it guides the user to unexplored areas in the scene upon request. In contrast to most guided tours, the user is free to follow suggestions only as much and as often as they are needed. Furthermore, the approach does not require any manual input or human preparation, as all necessary information is computed automatically.

The rest of the paper is structured as follows. In Section 2, we discuss relevant previous work. Afterward, our approach is introduced

*e-mail: {freitag | weyers | kuhlen}@vr.rwth-aachen.de

in Section 3. The pilot study and main user study we conducted to evaluate the technique are described in Section 4 and Section 5, before the study results are reported in Section 6 and discussed in Section 7. Finally, we discuss some limitations of the approach in Section 8 and give a conclusion and outlook on future work in Section 9.

2 RELATED WORK

A number of previous works have proposed interfaces to aid the exploration of a virtual environment (VE) or wayfinding in general. For example, basic orientation support can be provided by simple visual (e.g., [26]) or auditory (e.g., [10]) orientation indicators, or compass-like widgets that point to some locations of interest [5] or reference points [9]. Furthermore, overviews of the scene and the user's location within to improve wayfinding can be provided by virtual 2D maps [8, 9], or 3D World-in-Miniatures [44]. Moreover, it has been shown that highlighting landmarks in open environments [32], or selectively employing semi-transparency (see-through walls) in closed scenes [7] can improve wayfinding performance. In addition, to highlight which parts of the environment the user has already visited, manually placed breadcrumbs [10] or automatically drawn footprints or trails in the VE [9, 20, 35] or on maps [9] have been proposed. Trails have been shown to improve exploration efficiency, but clutter the environment in later visits [35]. However, while benefits of many of these general wayfinding aids for exploration have been demonstrated, they do not actively ensure, e.g., that the user does not miss important parts of the scene during exploration.

This is addressed by guided exploration techniques. For example, to ensure effective and efficient exploration, the automated computation of a virtual tour through selected locations of interest has been proposed [6, 12]. However, although the tour is computed automatically, this approach requires all interesting positions to be specified manually for each scene, which can be undesirable. Therefore, to identify relevant areas automatically, the use of *viewpoint quality* has been proposed, which assigns each position in a scene a quality value, usually based on an estimation of the amount of information that can be gained about the scene from that position [34,36,38,48]. Viewpoint quality can be computed in a variety of ways (see, e.g., [11, 15, 34, 36] for overviews of different approaches) and for different kinds of scene representations, such as geometry-based scenes (e.g., [33, 36, 48]), volume data (e.g., [4, 51]), and point clouds (e.g., [21]). Furthermore, while most approaches determine the quality completely automatically, some incorporate some user input, e.g., on the importance of certain scene entities (e.g., [39, 40, 51]). For virtual tours, viewpoint quality has been used to determine a favorable start position (e.g., [2]), continuously move the user towards unseen high-quality locations (e.g., [48, 50]), determine a set of best or representative positions as waypoints for a camera path (e.g., [24, 42, 46, 53]) or multiple paths [40], and to identify the most relevant areas in a scene [1], and can also be used to automatically control the travel speed [16]. In our work, viewpoint quality is also used to identify interesting target locations.

To enable some interactivity while following virtual tours, the *river analogy*, which allows for some deviations from the camera path, has been proposed [13, 19], as well as systems that enable some interactive control of movement direction and speed [29]. Moreover, to further increase interactivity, it is possible to only make suggestions for interesting locations that users can ignore if they are not interested [52].

However, none of these virtual tour approaches allow for the user to explore parts of the scene on their own, and to only use automated support to identify regions they have missed. Instead, either at least a part of the static tour has to be followed at least roughly, or suggestions can be followed that do not take into account whether the user has already visited the recommended locations. In contrast, our approach can be used to automatically generate suggestions to interesting, but yet unexplored regions.

Our approach is based on an idea that has been suggested previously [18], but has never been completely implemented, examined or evaluated. It builds upon an analysis of visibility information in areas throughout the scene reachable using ground-based navigation interfaces [18]. While the basic approach of determining which entities a user or agent has already seen has been previously employed to compute sets of representative viewpoints (e.g., [24, 39, 40, 48, 49]) and virtual tours (e.g., [24, 48, 50]), only binary visibility has been examined (i.e., it is not considered how well an object is visible or whether it has been observed shortly or examined extensively), it has not been used interactively, nor to support free exploration.

3 INTERACTIVE EXPLORATION ASSISTANCE INTERFACE

In essence, our approach is based on keeping track of which parts of the scene the user has already seen. Upon request, three suggestions for locations the user likely wants to visit are computed, based on viewpoint quality and taking into account what the user has already seen. For each target location, the user is provided with a "photo" taken at that position (Fig. 1, left). Furthermore, a suggested path to each target is visualized (Fig. 1, right). The paths are also computed considering both viewpoint quality and previously unseen regions, such that they prefer previously unknown connections, allowing the user to explore new parts of the environment also on their way to the target.

We utilize the visibility computation approach from [18], which determines omnidirectional visibility information in all *navigable* parts of the scene, defined by a *navigation mesh*. Visibility information is analyzed on object level and stored in a *visibility histogram*, i.e., for each position, the relative *visual size* of each object (the apparent size of the object to an observer at that position, equal to the solid angle subtended by it relative to the unit sphere) is stored. However, visibility information is only computed above the vertices of the navigation mesh, as these tend to be placed at positions where visibility changes (e.g., near corners or obstacles), and only interpolated in between. Furthermore, to ensure sufficient approximation accuracy, the navigation mesh is refined until adjacent visibility histograms are sufficiently similar.

By restricting the visibility analysis to navigable areas and by using adaptive refinement, the computation is usually completed in a short preprocessing step. Furthermore, the resulting visibility histograms can directly be used as input for viewpoint quality estimators. In addition, its underlying navigation mesh structure can directly be utilized to compute paths that consider both visibility and viewpoint quality.

3.1 Viewpoint Quality Adaptation Over Time

Any viewpoint quality metric based on objects can be used for our approach (cf., e.g., [15, 24, 39, 51]). However, to allow the quality metric to consider how well each object has already been examined, it is augmented by adding an *importance measure* or *weight* for each object (if not already part of the metric, cf. [39, 51]), with a weight of 1 representing an entirely unseen object, and a weight of 0 corresponding to a thoroughly examined (and thus uninteresting) object. While an object is visible to the user, its weight is steadily reduced, such that the viewpoint quality of all positions from where objects are visible that the user sees decreases over time. For many viewpoint quality schemes, the quality q for some position can thus be formulated as follows:

$$q = \sum_{i=1}^{N} w_i \cdot C(i), \qquad (1)$$

where N is the number of objects in the scene, w_i the weight of the i-th object, and $C(i)$ the contribution of the i-th object to the total viewpoint quality (depending on the metric used; usually, $C(i) = 0$ for invisible objects).

For our implementation, we use the *object uniqueness* metric [15], which awards higher quality scores to positions observing more objects that are *unique* in the scene. In this metric, $C(i) = \sqrt{\alpha_i \cdot U_i}$, where α_i is the visual size of the i-th object (how large it appears to the observer) and U_i is its relative uniqueness (inversely proportional to how many similar objects there are). Note, however, that our approach does not fundamentally depend on the concrete viewpoint quality metric, such that it can be exchanged to best fit the desired use case.

The weight reduction rate for each object depends on how well it is visible (how large it appears to the user) at each moment. This is expressed by its *visual size*, where 1 represents an object that completely surrounds the user, a size of 0 indicates an invisible object, and, e.g., 0.001 corresponds to about the size of a postcard at a distance of 1 m [18]. Furthermore, we define the minimum and maximum time it should take a visible object's weight to decrease to 0, with the minimum time t_{min} applying to "well visible" objects that have a visual size of at least vis_{max} (to prevent weights from being reduced arbitrarily fast) and the maximum time t_{max} representing the longest time it should take to diminish even the weight of small or distant objects. Thus, t_{max} is also the time it takes for a location to be considered "fully explored", i.e., the weight of all visible objects is 0, and thus its viewpoint quality becomes 0. In between, the time is simply interpolated linearly, i.e., the time it takes the weight of an object to decrease from 1 to 0, based on its visual size vis, is

$$t(vis) = \min\left(t_{\text{max}}, \frac{vis_{\text{max}} \cdot t_{\text{min}}}{\min(vis, vis_{\text{max}})}\right), \qquad (2)$$

such that in each second, each object's weight is reduced by $1/t(vis)$. In our implementation, we set $t_{\text{min}} = 5$ s, $t_{\text{max}} = 15$ s and $vis_{\text{max}} = 0.001$, but assert that these values only determine how quickly a location is considered explored (and therefore control how long a user is expected to examine each part of the scene), such that the approach should be robust to the choice of these values across scenes. Note that we do not distinguish objects that are partly occluded or farther away, if both have the same visual size. However, while it is easy to also compute, e.g., the ratio of visible surface area of each object [15], it is not clear how these cases should be treated differently. For example, while some objects need to be observed in their entirety, for many (especially simple) objects, seeing some part may be equally informative. Therefore, we decided not to artificially distinguish these cases, but assert them as interesting follow-up topics for future research.

Furthermore, in computing the weight decrease, we assume *omnidirectional visibility*, i.e., a 360° field of view for the user, which is not actually the case for human observers. Although our visibility analysis only supplies omnidirectional visibility directly, it is possible to estimate with reasonable accuracy if an object is actually in the user's field of view based on the object's center and the user's position and view direction. However, after conducting pilot tests with limited fields of view, we found that this requires users to actually look at all parts of the scene for an equal amount of time to reduce the viewpoint quality sufficiently. In contrast, we observed that users typically only glance at less interesting objects long enough to recognize them, while spending more time observing more interesting objects. Instead, using a restricted field of view would lead to the same locations being suggested repeatedly by the assistance interface until the user has looked sufficiently long even at less interesting objects.

3.2 Computation of Target Suggestions

When the user requests assistance, suggestions for three interesting and unexplored target location are generated—based on the current viewpoint quality everywhere in the scene—essentially by computing a *best set of views*. While computing such a set exactly is an NP-hard problem [31], in practice, it can often be approximated with a simple greedy approach (select the position with the highest viewpoint quality, consider the weight of all objects visible from there as 0, and repeat until enough positions are selected [18, 24, 48, 49]) or other global optimization strategies (cf., e.g., [14, 36, 39]). However, in exploration, we found that users often want to visit close locations first. Therefore, we select two informative but close locations, before adding the best remaining location. We follow the greedy approach mentioned above, first selecting the closest location whose viewpoint quality is better than 75% of all candidate locations (twice), then adding the globally best viewpoint at that time as the third suggestion. Furthermore, to avoid forcing the user to examine every object, only positions where at least 25% of visible objects retain a weight of at least 0.5 are considered as targets.

For each target suggestion, a path from the user's position is computed. To assist exploration, these paths are not only optimized to be short, but also to go through regions with a high viewpoint quality, which is found in both informative and unvisited areas. To compute paths that are navigable using ground-based travel interfaces, they are based on the navigation mesh also used to represent visibility. We use the A* algorithm [22] to determine the best path, using mid-points of navigation mesh edges as nodes and connections between all edges of the same navigation mesh polygon as edges [37], computing the travel cost for each edge $\{a,b\}$ inversely proportional to the square root of its viewpoint quality:

$$cost(a,b) = \max\left(\frac{q(a)+q(b)}{2}, 0.001 \cdot q_{\text{max}}\right)^{-0.5}, \qquad (3)$$

where $q(a)$ is the viewpoint quality at position a and q_{max} is the maximum viewpoint quality in the scene ignoring weights. The square root is used to mitigate the effect of large differences in viewpoint quality to prevent overly long detours around regions of low quality. Furthermore, using a viewpoint quality of at least $0.001 \cdot q_{\text{max}}$ ensures that the difference between the cost of traveling through the best and worst regions is limited even when the quality reaches 0 in completely explored areas (traveling through a completely explored region thus is about 30 times more costly than through the area with the highest viewpoint quality). The resulting paths are then simplified by successively deleting all nodes that can be removed without the path leaving the navigation mesh. Note that this step may increase path cost, but if there are several possible high-level ways to reach a target, the one chosen before will be maintained.

3.3 User Interface

We implemented the interface assuming two tracked hand controllers, and tried to make it as simple and easy to use as possible, as exploration assistance should impose as little cognitive load as possible on users [27]. When the user requests assistance by pressing the corresponding button on any of the controllers (we used the HTC Vive headset and controllers, and the "menu" button to request assistance), a "photo" of each target—a 110° wide-angle camera view from that location—with a colored frame is displayed above that controller (see Fig. 1, left). The "photos" stay attached to that controller, such that they can be moved out of the way or examined more closely by moving it. Furthermore, the distance to each target when following the suggested path is displayed in the lower right corner and updated while the user travels. In addition, arrows pointing up or down are placed above each "photo" whenever the corresponding target is more than 2 m above or below the user, to indicate that it is, e.g., on a different floor.

To show interesting and previously unseen parts of the scene, the camera orientation of each "photo" is chosen based on the viewpoint quality of its visible contents, which can be estimated from the omnidirectional visibility information by only taking into account visible objects whose center is located within the view frustum of the camera. We sample 36 camera directions (rotated around the vertical axis in 10° increments) and choose the direction with the highest viewpoint quality. The upright direction of each "photo" is kept identical to the global "up" direction to facilitate orientation in the picture.

Furthermore, at the same time, the suggested path to each target is visualized as a tube with the same color as the frame of the corresponding "photo", at 1.2 m above the ground. If several paths follow the same trajectory, they are placed side by side (see Fig. 1, right). In addition, the target position is visualized as a colored sphere.

When the user reaches a target or leaves a path (i.e., the distance to any point on the path is more than 5 m), it disappears along with the corresponding "photo" to reduce clutter, assuming that the user is no longer interested in this path. Furthermore, the user can select a target to make the other visualizations disappear by moving the other controller into one of the "photos" and pressing the trigger button.

All computations are performed in a separate thread to avoid interfering with the rendering thread, showing an activity indicator above the controllers as feedback. In all of our test scenes, the computation always takes less than 0.6 seconds. The interface and all runtime computations were implemented in Unity 2017.1, the visibility analysis was performed using C++ and OpenGL.

4 Pilot Study

To collect feedback and fine-tune the assistance interface, and to obtain an estimate of the time necessary to explore the three scenes we used for the main study (see Sec. 5.1 and Fig. 2), we conducted a pilot study among nine participants (4 female, 5 male, mean age 27.0, SD=8.6, all with virtual reality experience). Each of them explored all three scenes at a comfortable pace and without time limit until they felt confident that they had seen everything and had a good mental impression of the scene. They used an HTC Vive and the same travel technique that we also used for the main study (short-distance teleportation, see Sec. 5.2). Afterward, they were asked to try out the assistance interface and give their opinion on any of its properties. Among others, this led us to display the path visualizations at about chest height instead of on the ground (as often done with trails, e.g., [35]), draw them transparently, display the target position, update the remaining distance to the target while the user travels, and simplify the arrows to only point up or down (instead of pointing at the target similar to [5]).

For the *dungeon* scene, we removed one participant as an outlier who took twice as much time as the others. On average, participants needed 10:58 minutes to explore the *university*, 5:17 minutes for the *dungeon*, and 5:08 minutes for the *office*. For the main study, we used 75% of these times to increase the difficulty (cf. Sec. 5.3) and rounded to half minutes for better comprehensibility, yielding 8 minutes for the *university* and 4 minutes each for the *dungeon* and *office* scenes.

5 User Study

To evaluate whether the proposed interface actually conveys a benefit to users, we conducted a user study, asking participants to explore three different virtual environments. We chose a between-participants design, where one group (the experimental group) had access to the assistance interface, while the other (control group) did not. Exploration success was measured based on how well participants could answer questions about the scenes afterward, and by how many areas of the scene they visited and how many objects they saw.

Note that a comparison with existing assistance interfaces (such as trails or compasses) would also be beneficial for a complete evaluation. However, as there is no established standard for exploration assistance without manual preparation, we found a baseline comparison to be more helpful for a first evaluation of our approach. Nevertheless, we assert that an extended evaluation will be important in future work.

5.1 Scenes

To examine exploration success in different environments, we used three different scenes: a *university* building, a *dungeon* scene, and an *office*. All of them contained realistically modeled and rendered buildings, each featuring a different number and layout of rooms and corridors. Furthermore, all scenes were static and did not allow any interaction. Overviews of each scene can be found in Figure 2.

The first scene is a large, open, and uncluttered *university* building, surrounded by a sparse outside area (Fig. 2, left). The building has a large two-story lecture hall and two floors containing offices, seminar rooms, bathrooms, and a terrace. Its total area is approximately 4,700 m^2, with an additional 13,300 m^2 outside area. The second scene is a smaller, dimly lit *dungeon*, featuring a number of small rooms and corridors, most of which are located in the middle one of its three stories (Fig. 2, center). The upper story only contains a large entrance hall (where the scene was entered), while the lower story consists of only two rooms. In total, the dungeon covers about 292m^2 (excluding the entrance hall). The third scene is an abandoned *office* environment with a simple structure, consisting of a corridor, a number of rooms, and a staircase to an upper floor that contains only one large office with a small anteroom (Fig. 2, right). Its total area is about 255m^2.

The navigation meshes for the visibility analysis were computed using the open-source toolset *Recast*[1] and pruned to the largest connected component. We used the visibility analysis from [18] with a refinement threshold of $\theta = 0.8$, which took 67.1 s for the *university* (7912 visibility samples), 5.5 s for the *dungeon* (730 samples), and 4.1 s for the *office* scene (561 samples) using an Intel Xeon E5-1603 2.8 GHz CPU and an NVIDIA GeForce GTX 1070 GPU.

5.2 Interface and Hardware

We used an HTC Vive virtual reality headset to display the virtual environments. Participants carried a Vive controller in each hand and could physically move within an area of 3.5 m$\times$3.5 m. For further navigation, we decided to rely on short-distance teleportation to prevent cybersickness, despite a possible negative impact on spatial orientation. We used the teleportation implementation provided by the SteamVR asset for Unity, which we configured to allow short-distance movement of up to 7 m at a time by pressing and holding any controller thumb pad while pointing at the target location, and to fade to black in between. Participants in the experimental group could additionally use the menu button on any controller to call up the assistance interface, and the trigger button to select a target, as described in Section 3. During the study, a countdown showing the remaining time was displayed on the virtual representation of each controller. The environments were rendered using Unity 2017.1, maintaining a constant frame rate of 90 Hz.

5.3 Procedure

Participants started by giving their informed consent and filled out both a demographic questionnaire and Kennedy's Simulator Sickness Questionnaire (SSQ) [25]. Then, they received written instructions about the use of the virtual reality system and their task. Participants in the experimental group additionally got instructions on how to use the assistance interface.

Afterward, they put on the Vive headset and started with a training scene in a virtual bookstore (consisting of several rooms and two stories) to get acquainted with the hardware, physical walking and teleportation, and (in the experimental group) the assistance interface until they felt comfortable using them. Subsequently, they completed the study task by exploring the three scenes, in an order counterbalanced using a Latin square design, having 8 minutes for the *university*, and 4 minutes for each of the other scenes (cf. Sec. 4). They were told to "explore each scene completely", and that afterward, different questions would be asked about each scene. Participants with the assistance interface were not restricted in how and when to use it (e.g., to guide their complete exploration, to only find locations they had missed, or not at all). After each scene, they could take a short break if they wanted. After the third scene, participants took off the headset, answered Kennedy's SSQ again as well as the NASA-TLX subjective task load questionnaire [23], and a Likert-scale questionnaire asking for each scene how confident participants felt of having explored it completely, and whether they felt that there had been sufficient time.

Then, they answered a questionnaire posing a series of questions about each of the scenes, partitioned into three broad categories. *Survey* or *structural* questions aimed at testing participants' survey knowledge of the scene, e.g., by asking on which side of the building a certain room was located, *route* questions tested route knowledge, by asking how to get from one place to another, and *detail* questions aimed at revealing participants' knowledge of details, e.g., by asking which of a list of objects existed in the scene, or where a certain object

[1] https://github.com/recastnavigation/recastnavigation

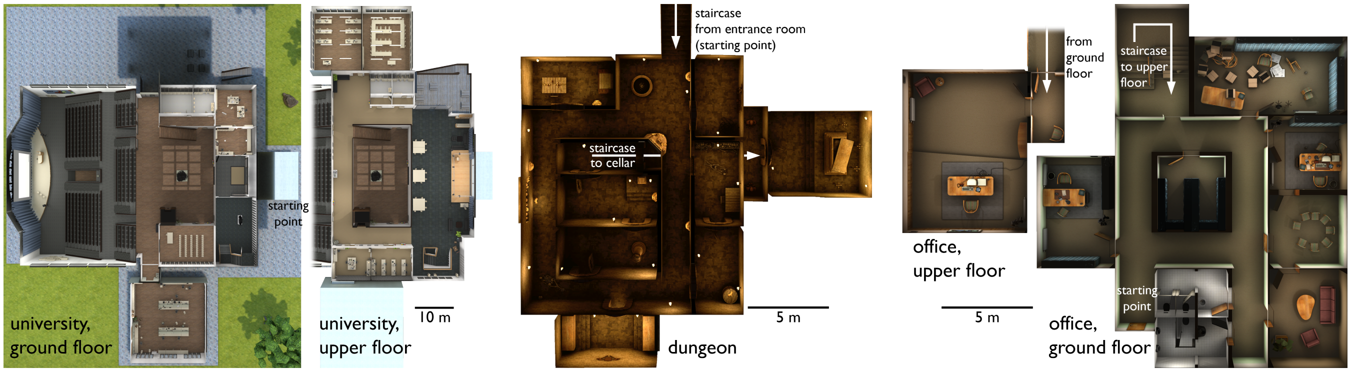


Figure 2: Overviews of the three scenes used in the study.

was located. In total, there were 35 questions (16 for the *university*, 9 for the *dungeon*, and 10 for the *office* scene). Note that these were not all questions that could have been asked, but were designed to cover all parts of each scene approximately equally, while limiting the number of questions to reduce the time needed to complete the questionnaire. Furthermore, while it would have been easier for participants to answer the questionnaire for each scene directly after they had explored it, we decided against this for organizational reasons.

Finally, participants answered the SUS presence questionnaire [47] and a concluding questionnaire about the assistance interface (in the experimental group). The total procedure took 75.6 minutes on average (SD=11.8 min.), of which about 21 minutes were spent wearing the headset (5 min. for training and 16 min. exploring the scenes).

5.4 Hypotheses

The assistance interface guides users to regions they have not found on their own. Therefore, we assume that it helps users learn and remember more about each scene (leading to **H1**). Furthermore, we expect users to explore the scenes more *completely*, i.e., miss fewer places (leading to **H2** and **H3**). In addition, we suspect the fact that the interface informs users of places left to explore to increase their *confidence* of having explored each scene completely (leading to **H4**). Thus, our hypotheses about participants with the assistance interface are:

H1 They will perform better answering questions about the scenes.
H2 They will visit more rooms/areas of each scene on average.
H3 They will see more objects contained in each scene.
H4 They will be more confident of their exploration success.

5.5 Participants

In total, 54 individuals (9 female, 45 male, mean age 26.6 years, SD=6.1) participated in and completed the experiment, none of whom had taken part in the pilot study or had any knowledge of the scenes used in the study. 29 participants had previously experienced virtual reality for at least 10 minutes, either as the only user of a CAVE or using a tracked head-mounted display. Furthermore, 45 participants reported that they regularly played video games or had done so in the past, and 46 stated that they usually had good orientation skills. Note that we did not perform a standardized orientation test to avoid increasing the duration of the study further, but simply asked participants whether their orientation skills were usually good, which might reduce the reliability of the answer.

Participants were assigned to each of the two study groups in an alternating fashion, counter-balancing for gender, orientation skills, and whether they had virtual reality or video game experience, such that the number of participants with any of these properties varied by at most one between groups. To incentivize participants to make an effort to explore successfully and to answer the questions, prizes valuing 80€ in total were distributed among the two participants of each group who had reached the highest score in answering the questions.

Table 1: Ratings of participants' confidence of having explored each scene completely, and if there was sufficient time to do so, on a scale of 1="not at all" to 7="completely". Significant differences are highlighted.

	with assistance	control group	*p*
confidence overall	5.89	5.52	0.198
university	5.15	5.26	0.777
dungeon	6.07	5.67	0.285
office	6.44	5.63	**0.026**
sufficient time overall	5.72	5.36	0.233
university	4.85	4.88	0.935
dungeon	6.00	5.73	0.471
office	6.31	5.46	**0.043**

6 Results

We analyzed the results using independent-samples t tests (interval data), Wilcoxon signed-rank tests (Likert scales), and Mann-Whitney U tests (binary data), at the 0.05 level of significance. For all independent-samples t tests, we used Welch-Satterthwaite adjustments to the degrees of freedom whenever Levene's test indicated that the assumption of equality of variances was violated.

The results of the SUS presence and NASA-TLX task load questionnaires, as well as the responses to whether participants felt confident they had explored each scene completely and whether there was sufficient time (on a scale of 1="not at all" to 7="completely") were analyzed using independent-samples t tests. There was no significant difference in perceived task load between conditions (with assistance: mean 32.8, SD=13.2, without: 30.1, SD=13.4, on a scale of 0 to 100, p=0.449), nor in presence (with assistance: mean 4.85, SD=1.03, without: 4.89, SD=0.99, p=0.876). Averaged over all scenes, participants were highly confident that they had explored each scene completely (mean 5.7 on a scale of 1 to 7), and felt that there had been sufficient time (mean 5.5), but differences between groups were only significant for the *office* scene (see Tab. 1). Average SSQ scores were 9.42 (SD=11.73) before and 14.96 (SD=11.35) after the experiment, indicating an overall low incidence of simulator sickness with a small increase during the experiment. The increase in SSQ between groups was not significantly different (mean for both: +5.54, p=1.000).

In evaluating the questionnaire about the scenes, we gave 1 point for each correctly answered question. In multiple-choice questions (such as "Which of the following objects existed in the scene?") with n correct answers, we awarded $1/n$ points for each correctly checked answer and deducted $1/n$ points for each incorrectly checked answer (however, the score for each question could not go below 0). The number of questions for each scene, as well as the number of questions in each category (survey/structural, route, details) varied between scenes, as we aimed at representing all parts of the scene equally rather than asking the same number of questions. Therefore, we weighted

Table 2: Scores for the questionnaire about the three scenes. The values of individual questions are weighted by the time participants spent in each scene, and each category (survey, route, details) was weighted to represent one third of each scene's score. The unweighted scores (equal weight for each question) are given as well for reference.

	with assistance	control group	p
total score	51.9%	41.8%	**0.012**
university	51.6%	42.1%	**0.031**
dungeon	50.1%	36.6%	**0.027**
office	54.3%	46.2%	0.150
total score (unweighted)	52.4%	43.6%	**0.015**
university (unweighted)	52.7%	43.8%	**0.026**
dungeon (unweighted)	48.4%	35.0%	**0.018**
office (unweighted)	55.4%	50.8%	0.364
survey/structural questions	57.2%	47.6%	**0.031**
route questions	50.4%	32.9%	**0.012**
detail questions	48.1%	44.8%	0.302

Table 3: Average number of areas and rooms (as opposed to corridors, staircases or outside area) participants missed in each scene.

	with assistance	control group	p
university – all areas	2.5 (10.3%)	4.4 (18.4%)	**0.001**
university – rooms	0.8 (5.8%)	3.1 (22.2%)	**0.000**
dungeon – all areas	0.5 (3.4%)	3.7 (26.7%)	**0.000**
dungeon – rooms	0.5 (4.8%)	3.1 (31.1%)	**0.000**
office – all areas	0.8 (6.8%)	1.6 (13.0%)	**0.047**
office – rooms	0.8 (8.0%)	1.4 (14.0%)	0.063

the scores for each scene by the time participants spent exploring it, such that the *university* accounted for 50% and each of the other scenes for 25% of the total score. Furthermore, we weighted each category of questions to account for $1/3$ of the score for each scene. The results are summarized in Table 2, which also provides unweighted scores to show that the overall results are independent of the weighting.

Furthermore, we evaluated how *complete* participants explored each scene by measuring how many parts of each scene they visited. To prepare for this, we partitioned each scene into *areas*, each comprising a room, corridor or staircase, or, in the case of the *university*, also the foyer and outside area. We then compared how many of these areas participants visited or missed in each condition, especially focusing on rooms, as most information about the scenes can arguably be gathered by visiting rooms. The results are summarized in Table 3. Closer inspection of the rooms that were missed reveals that in the *university*, participants with assistance interface visited two of the three seminar rooms significantly more often (each visited by 24–25 of the 27 participants with assistance, but only by 18–19 without, p=0.019, 0.037 and 0.052, determined using Mann-Whitney U tests), as well as three of the four bathrooms (24–25 with assistance, 11–18 without, $p<0.019$), but spent significantly less time in the sparse outside area (46.8 s with assistance, 82.5 s without, p=0.001). In the *dungeon*, participants with assistance visited all of the central 4 rooms (including the cellar) significantly more often (each visited by 25–27 with assistance, but only 8–11 without, $p<0.001$). In the *office*, participants with assistance visited the upper floor significantly more often (all 27 participants with assistance, but only 21 without, p=0.010).

In addition, we assessed the completeness of the exploration by looking at how many objects participants had seen. For this, we evaluated the average number of objects with a weight of less than 1 (cf. Sec. 3) at the end of the exploration. The results are summarized in Table 4.

Furthermore, participants with assistance rated items regarding the interface on a 5-point Likert scale. We tested if the median response differed from the neutral value of 3 ("undecided") using one-sample Wilcoxon signed-rank tests. The results are summarized in Table 5.

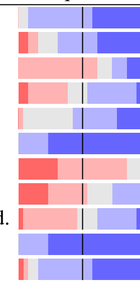
As we did not find different effects when analyzing the results

Table 4: Average number of objects seen in each scene (assuming omnidirectional visibility).

	with assistance	control group	p
university	1909	1845	**0.010**
dungeon	324	278	**0.000**
office	534	505	**0.030**

Table 5: Results of the questionnaire regarding the assistance interface.

	Frequencies	Med.	p
The interface was helpful.		4	**0.000**
Without it, I would have missed something.		4	**0.006**
It helped understanding the scene's structure.		2	0.472
It helped finding details in the scene.		3	0.741
The photos' perspective was chosen well.		4	**0.001**
It was easy to follow the paths.		5	**0.000**
The visualizations were annoying/distracting.		2	**0.000**
They made the environments less realistic.		2	0.057
While following a path, I hardly looked around.		3	0.874
The interface was easy to understand and use.		5	**0.000**
I would use it again to explore other places.		4	**0.000**



1 strongly disagree 2 disagree 3 undecided 4 agree 5 strongly agree

based on participants' virtual reality experience, details are omitted for brevity. We did not analyze the data based on gender, video game experience, or orientation skills due to the limited number of female participants, participants without video game experience, or with (self-assessed) bad orientation skills.

7 Discussion

Participants with the assistance interface could correctly answer more questions about two of the three scenes and in two of three categories, largely confirming hypothesis **H1**. Furthermore, they visited more areas in each scene, and saw more objects of each scene, confirming hypotheses **H2** and **H3**. Although we did not formally evaluate how the assistance interface was used, we informally observed a wide variety of strategies, with no clear predominant pattern. While some participants only used it after exploring most of the scene on their own, others relied on it from the beginning. Furthermore, while many tended to follow the paths to the target, others only used them as guidance, leaving the path when they encountered an interesting location mid-way.

Evidently, the reason for the more complete exploration (cf. Tab. 3 and 4) is that participants used the assistance interface to find locations they had not visited yet. In contrast, participants in the control group tended to miss parts of the environment in all of the three scenes, an issue which therefore seems to occur independently of some scene properties. In the *university*, which is an open, clear and uncluttered, but also quite large scene, participants without assistance probably just lost track of where they had already been and which rooms existed. In contrast, in the *dungeon* scene, which is a smaller, but dark and somewhat obscure place, many participants simply seemed to have overlooked the entrance to the central part of the scene. The results from the *office*, which is a small and well-lit place, but where still 6 of 27 participants in the control group missed the upper floor, illustrate that without assistance, users explore incompletely even without complicating factors such as size or obscurity. In contrast, in all three scenes, the assistance interface seems to have prevented most participants from missing parts of the environment for any of these reasons.

Apparently, this is also the reason why participants achieved higher scores answering questions about the *university* and *dungeon* scenes. While there were also two questions about the *office* that could not be answered correctly without having seen the upper floor, one of them was apparently too hard (so even participants who had seen the upper floor often answered incorrectly) and the other seems to have been misunderstood frequently, such that their effect on the score was limited.

Considering the question categories, there were significant benefits of the assistance interface for questions regarding survey and route knowledge, but not concerning details (cf. Tab. 2). Apparently, the improved survey knowledge stems from just having explored the scenes more completely. Concerning route knowledge, while it is possible that it was improved by following the suggested paths, these probably only rarely coincided with a route asked in the questionnaire. More likely, it could be an effect carried over from improved survey knowledge. Although none of the start or target positions of any of the route knowledge questions were in areas missed significantly less often by participants with assistance, users with an overall better cognitive map of the environment can also be expected to be able to describe routes more correctly. Note, however, that we only asked seven route questions in total to limit the time needed to answer the questionnaire.

Regarding detail questions, there was no significant difference between conditions. However, as the main benefit of the assistance interface appears to stem from not missing parts of the scene, the lack of an advantage in this category seems to be due to the fact that only very few detail questions concerned areas that were often missed.

Contrary to **H4**, participants with the assistance interface only seemed to be more confident of their exploration success in the *office* scene (cf. Tab. 1). While not too much should be concluded from non-significant results, we expected confidence to increase mainly due to the interface confirming that no interesting places have been missed (cf. Sec. 5.4). However, contrary to the *office*, where most participants could exhaust all suggestions, in the *university*, only few users managed to do so within the 8-minute time limit. Furthermore, in the *dungeon*, although most users with assistance visited all suggestions, participants in the control condition who missed the central part of the scene spent much more time on the remaining areas, likely also increasing their (unjustified) confidence of a successful exploration.

The results of the questionnaire regarding the assistance interface (cf. Tab. 5) indicate that participants found it helpful as well as easy to understand and use. Furthermore, the visualizations did not seem to get in the way, and the choice of the "photos" of the target locations (with the target placed at a position with high viewpoint quality, facing in a direction that had high viewpoint quality) was helpful. These results affirm our original goal of creating a useful assistance interface that does not increase the user's cognitive load.

However, the study design did have some limitations. First, the study task was not entirely identical to how users normally explore a virtual environment, which is often an only semi-conscious process, and usually not strictly limited in time. While it would have been possible to give users a different main task (such as a search task), we expect that this would have steered user behavior in some specific direction, while at the same time leading to vastly different exploration results if, e.g., building a good cognitive map was not strictly necessary for that task. Furthermore, the available time was rated as sufficient by most participants (cf. Tab. 1), although allowing a little more time for the exploration of the *university* would probably have been beneficial. In total, we are confident that the results should generalize well to the actual, free exploration of an unknown virtual environment.

Second, we found that some of the questions about the scenes were often misunderstood, increasing the level of noise in the results. For example, we asked how many offices had existed in the *office* scene. While we did not consider the chaotic room (upper right in Fig. 2, right) to be an office, some participants apparently thought so (as it contained a desk), while some others seemed to consider *every* room an office. This underlines that questions like this should be carefully evaluated for possible misunderstandings, e.g., in a pilot study. In addition, it would have been possible to have participants sketch maps of the environments (cf. [3,17,45]). However, we decided against this, as we regarded the scenes as too complex for simple sketches, and did not want to increase the already substantial time necessary to answer the questionnaires even further. Nevertheless, as none of these arguments apply differently to any study group, we think that overall, the questions sufficed to compare the exploration success between groups.

Third, we only examined (mainly) indoor, realistic architectural scenes explored using a ground-based travel interface. While we are confident that the results should generalize to outdoor scenes, it is unclear if the interface would also be beneficial, e.g., in abstract scenes.

8 Limitations of the Method

Our approach builds upon the visibility analysis for navigable surfaces proposed in [18]. While this is ideal for ground-based exploration (which is an important case especially for realistic scenes), as the visibility preprocessing can be completed very efficiently, it prevents the direct application of the method to arbitrary travel interfaces. Although the visibility analysis technique could be exchanged (e.g., by extending the method from [18] to 3D, or the approach from [30] to quantitative instead of binary visibility) if the increased resources required for such an analysis are available or the scene is small, it is not clear whether following a path in three dimensions would be as natural and lead to the same positive results.

Furthermore, in its current implementation, the proposed approach can only be used for one-time exploration—when an area is completely explored, it is never going to be suggested as a target again. However, it might be beneficial to have the option to be guided to the least explored (but still interesting) locations later on to revisit them. This could be realized by replacing the weight reduction function by, e.g., an exponential decrease, such that the object weights are never completely reduced to zero, coupled with an adaptive relaxation of the requirements for a position to be eligible as target suggestion (at the moment, at least 25% of visible objects have to retain a weight of at least 0.5). Alternatively, the interface could switch to this option when the scene has been explored completely, or upon user command.

Finally, the assistance interface always recommends the complete exploration of the scene. While less interesting regions (with lower viewpoint quality) are suggested last, the user still has to go there to be sure that there is no further, possibly interesting location that may have received a lower quality score. E.g., in the study, participants tended to examine the bathrooms only very briefly, such that the weights of objects there were not sufficiently decreased and they were often suggested again at the end. However, this could be solved by an option either to display an overview of more than three recommendations—to ensure that no interesting area classified as less informative remains—or to interactively discard locations from a distance by reducing the weight of objects visible from there upon command.

9 Conclusion

We have presented an interactive assistance interface to aid the immersive exploration of virtual environments using ground-based travel interfaces. Based on an analysis of what the user has already seen and which further locations are informative according to viewpoint quality, it suggests interesting targets and visualizes paths leading there that likewise visit interesting regions. In a user study, we found that the technique improves the knowledge of the scene, leads to a more complete exploration, and is experienced as helpful and easy to use.

In future work, we want to examine extensions of the interface, especially regarding the re-exploration of less thoroughly visited regions of a scene. Furthermore, we want to test the approach in further types of virtual environments, including actual outdoor scenes, but also abstract scenes and visualizations.

References

[1] C. Andújar, P. Vázquez, and M. Fairén. Way-Finder: Guided Tours through Complex Walkthrough Models. *Computer Graphics Forum*, 23(3):499–508, 2004.

[2] P. Barral, G. Dorme, and D. Plemenos. Intelligent Scene Exploration with a Camera. In *Int. Conf. 3IA*, 2000.

[3] M. Billinghurst and S. Weghorst. The Use of Sketch Maps to Measure Cognitive Maps of Virtual Environments. In *Virtual Reality Annual International Symp.*, pp. 40–47, 1995.

[4] U. D. Bordoloi and H.-W. Shen. View Selection for Volume Rendering. In *IEEE Visualization*, pp. 487–494, 2005.

[5] L. Chittaro and S. Burigat. 3D Location-Pointing as a Navigation Aid in Virtual Environments. In *Working Conference on Advanced Visual Interfaces*, pp. 267–274, 2004.

[6] L. Chittaro, R. Ranon, and L. Ieronutti. Guiding Visitors of Web3D Worlds through Automatically Generated Tours. In *8th Int. Conf. on 3D Web Technology*, pp. 27–38, 2003.

[7] L. Chittaro and I. Scagnetto. Is Semitransparency Useful for Navigating Virtual Environments? In *ACM Symp. on Virtual Reality Software and Technology*, pp. 159–166, 2001.

[8] R. P. Darken and H. Cevik. Map Usage in Virtual Environments: Orientation Issues. In *IEEE Virtual Reality Conference (VR)*, pp. 133–140, 1999.

[9] R. P. Darken and B. Peterson. Spatial Orientation, Wayfinding, and Representation. In *K. M. Stanney (Ed.), Handbook of Virtual Environments: Design, Implementation, and Applications*, pp. 493–518, 2001.

[10] R. P. Darken and J. L. Sibert. A Toolset for Navigation in Virtual Environments. In *ACM Symp. on User Interface Software and Technology*, pp. 157–165, 1993.

[11] H. Dutagaci, C. P. Cheung, and A. Godil. A Benchmark for Best View Selection of 3D Objects. In *ACM Workshop on 3D Object Retrieval*, pp. 45–50, 2010.

[12] N. Elmqvist, M. E. Tudoreanu, and P. Tsigas. Tour Generation for Exploration of 3D Virtual Environments. In *ACM Symp. on Virtual Reality Software and Technology*, pp. 207–210, 2007.

[13] N. Elmqvist, M. E. Tudoreanu, and P. Tsigas. Evaluating Motion Constraints for 3D Wayfinding in Immersive and Desktop Virtual Environments. In *SIGCHI Conference on Human Factors in Computing Systems*, pp. 1769–1778, 2008.

[14] M. Feixas, M. Sbert, and F. González. A Unified Information-Theoretic Framework for Viewpoint Selection and Mesh Saliency. *ACM Trans. on Applied Perception (TAP)*, 6(1):1:1–1:23, 2009.

[15] S. Freitag, B. Weyers, A. Bönsch, and T. W. Kuhlen. Comparison and Evaluation of Viewpoint Quality Estimation Algorithms for Immersive Virtual Environments. In *ICAT-EGVE*, pp. 53–60, 2015.

[16] S. Freitag, B. Weyers, and T. W. Kuhlen. Automatic Speed Adjustment for Travel through Immersive Virtual Environments based on Viewpoint Quality. In *IEEE Symp. on 3D User Interfaces (3DUI)*, pp. 67–70, 2016.

[17] S. Freitag, B. Weyers, and T. W. Kuhlen. Examining Rotation Gain in CAVE-like Virtual Environments. *IEEE Trans. on Visualization and Computer Graphics*, 22(4):1462–1471, 2016.

[18] S. Freitag, B. Weyers, and T. W. Kuhlen. Efficient Approximate Computation of Scene Visibility Based on Navigation Meshes and Applications for Navigation and Scene Analysis. In *IEEE Symp. on 3D User Interfaces (3DUI)*, pp. 134–143, 2017.

[19] T. A. Galyean. Guided Navigation of Virtual Environments. In *Symp. on Interactive 3D Graphics*, pp. 103–104, 1995.

[20] D. Grammenos, M. Filou, P. Papadakos, and C. Stephanidis. Virtual Prints: Leaving Trails in Virtual Environments. In *Eurographics Symp. on Virtual Environments (EGVE)*, pp. 131–138, 2002.

[21] Z. Habibi, G. Caron, and E. M. Mouaddib. 3D Model Automatic Exploration: Smooth and Intelligent Virtual Camera Control. In *Asian Conference on Computer Vision*, pp. 612–626, 2014.

[22] P. E. Hart, N. J. Nilsson, and B. Raphael. A Formal Basis for the Heuristic Determination of Minimum Cost Paths. *IEEE Trans. on Systems Science and Cybernetics*, 4(2):100–107, 1968.

[23] S. G. Hart and L. E. Staveland. Development of NASA-TLX (Task Load Index): Results of Empirical and Theoretical Research. *Advances in Psychology*, 52:139–183, 1988.

[24] B. Jaubert, K. Tamine, and D. Plemenos. Techniques for Off-line Scene Exploration Using a Virtual Camera. In *Int. Conf. 3IA*, vol. 6, 2006.

[25] R. Kennedy, N. Lane, K. Berbaum, and M. Lilienthal. Simulator Sickness Questionnaire: An Enhanced Method for Quantifying Simulator Sickness. *The Int. Journal of Aviation Psychology*, 3(3):203–220, 1993.

[26] A. Khan, I. Mordatch, G. Fitzmaurice, J. Matejka, and G. Kurtenbach. ViewCube: A 3D Orientation Indicator and Controller. In *Symp. on Interactive 3D Graphics and Games*, pp. 17–25, 2008.

[27] J. J. LaViola Jr, E. Kruijff, R. P. McMahan, D. Bowman, and I. P. Poupyrev. *3D User Interfaces: Theory and Practice*. Addison-Wesley Professional, 2017.

[28] K. Lynch. *The Image of the City*. MIT Press, 1960.

[29] S. Mirhosseini, J. Marino, and A. E. Kaufman. Automatic Speed and Direction Control along Constrained Navigation Paths. In *IEEE Virtual Reality Conference (VR)*, pp. 29–36, 2017.

[30] S. Nirenstein and E. H. Blake. Hardware Accelerated Visibility Preprocessing using Adaptive Sampling. In *Eurographics Workshop on Rendering*, 2004.

[31] J. O'Rourke. *Art Gallery Theorems and Algorithms*, vol. 57. Oxford University Press, 1987.

[32] J. S. Pierce and R. Pausch. Navigation with Place Representations and Visible Landmarks. In *IEEE Virtual Reality Conference (VR)*, pp. 173–288, 2004.

[33] D. Plemenos and M. Benayada. Intelligent Display in Scene Modeling. New Techniques to Automatically Compute Good Views. In *Int. Conf. GraphiCon*, 1996.

[34] O. Polonsky, G. Patané, S. Biasotti, C. Gotsman, and M. Spagnuolo. What's in an image? *The Visual Computer*, 21(8-10):840–847, 2005.

[35] R. A. Ruddle. The Effect of Trails on First-Time and Subsequent Navigation in a Virtual Environment. In *IEEE Virtual Reality Conference (VR)*, pp. 115–122, 2005.

[36] M. Sbert, D. Plemenos, M. Feixas, and F. González. Viewpoint Quality: Measures and Applications. In *Eurographics Workshop on Computational Aesthetics in Graphics, Visualization and Imaging*, pp. 185–192, 2005.

[37] G. Snook. Simplified 3D Movement and Pathfinding using Navigation Meshes. *Game Programming Gems*, 1(1):288–304, 2000.

[38] D. Sokolov and D. Plemenos. Viewpoint Quality and Scene Understanding. In *Int. Conf. on Virtual Reality, Archaeology and Intelligent Cultural Heritage*, pp. 67–73, 2005.

[39] D. Sokolov and D. Plemenos. High Level Methods for Scene Exploration. *Journal of Virtual Reality and Broadcasting*, 3(12), 2006.

[40] D. Sokolov and D. Plemenos. Virtual World Explorations by using Topological and Semantic Knowledge. *The Visual Computer*, 24(3):173–185, 2008.

[41] D. Sokolov, D. Plemenos, and K. Tamine. Methods and Data Structures for Virtual World Exploration. *The Visual Computer*, 22(7):506–516, 2006.

[42] D. Sokolov, D. Plemenos, and K. Tamine. Viewpoint Quality and Global Scene Exploration Strategies. In *Int. Conf. on Computer Graphics Theory and Applications (GRAPP)*, vol. 2006, pp. 184–191, 2006.

[43] S. D. Steck and H. A. Mallot. The Role of Global and Local Landmarks in Virtual Environment Navigation. *Presence: Teleoperators and Virtual Environments*, 9(1):69–83, 2000.

[44] R. Stoakley, M. J. Conway, and R. Pausch. Virtual Reality on a WIM: Interactive Worlds in Miniature. In *SIGCHI Conference on Human Factors in Computing Systems*, pp. 265–272, 1995.

[45] E. Suma, S. Clark, D. Krum, S. Finkelstein, M. Bolas, and Z. Warte. Leveraging Change Blindness for Redirection in Virtual Environments. In *IEEE Virtual Reality Conference (VR)*, pp. 159–166, 2011.

[46] J. Tao, J. Ma, C. Wang, and C.-K. Shene. A Unified Approach to Streamline Selection and Viewpoint Selection for 3D Flow Visualization. *IEEE Trans. on Visualization and Computer Graphics*, 19(3):393–406, 2013.

[47] M. Usoh, E. Catena, S. Arman, and M. Slater. Using Presence Questionnaires in Reality. *Presence*, 9(5):497–503, 2000.

[48] P.-P. Vázquez, M. Feixas, M. Sbert, and W. Heidrich. Viewpoint Selection using Viewpoint Entropy. In *Vision Modeling and Visualization Conference (VMV)*, pp. 273–280, 2001.

[49] P.-P. Vázquez, M. Feixas, M. Sbert, and W. Heidrich. Automatic View Selection Using Viewpoint Entropy and its Application to Image-Based Modelling. *Computer Graphics Forum*, 22(4):689–700, 2003.

[50] P. P. Vázquez and M. Sbert. Automatic Indoor Scene Exploration. In *6th Int. Conf. Computer Graphics & Artificial Intelligence*, pp. 13–24, 2003.

[51] I. Viola, M. Feixas, M. Sbert, and M. E. Groller. Importance-Driven Focus of Attention. *IEEE Trans. on Visualization and Computer Graphics*, 12(5):933–940, 2006.

[52] E. A. Wernert and A. J. Hanson. A Framework for Assisted Exploration with Collaboration. In *IEEE Visualization*, pp. 241–529, 1999.

[53] J. Zhao, H. Song, J. Xu, and L. Wu. Automatic Exploration Path Planning for 3D Object Observation. In *Int. Joint Conf. on INC, IMS and IDC (NCM'09)*, pp. 1289–1294, 2009.