Hybrid Orbiting-to-Photos in 3D Reconstructed Visual Reality

Benjamin Nuernberger
University of California, Santa Barbara
Santa Barbara, California
benjamin.nuernberger@gmail.com

Tobias Höllerer
University of California, Santa Barbara
Santa Barbara, California
holl@cs.ucsb.edu

Matthew Turk
University of California, Santa Barbara
Santa Barbara, California
mturk@cs.ucsb.edu

ABSTRACT

Virtually navigating through photos from a 3D image-based reconstruction has recently become very popular in many applications. In this paper, we consider a particular virtual travel maneuver that is important for this type of virtual navigation—orbiting to photos that can see a point-of-interest (POI). The main challenge with this particular type of orbiting is how to give appropriate feedback to the user regarding the existence and information of each photo in 3D while allowing the user to manipulate three degrees-of-freedom (DoF) for orbiting around the POI. We present a hybrid approach that combines features from two baselines—proxy plane and thumbnail approaches. Experimental results indicate that users rated our hybrid approach more favorably for several qualitative questionnaire statements, and that the hybrid approach is preferred over both baselines for outdoor scenes.

CCS CONCEPTS

• **Human-centered computing** → *Virtual reality*; • **Computing methodologies** → *Graphics systems and interfaces*;

KEYWORDS

virtual navigation, orbiting, 3D reconstructed visual reality

ACM Reference Format:
Benjamin Nuernberger, Tobias Höllerer, and Matthew Turk. 2018. Hybrid Orbiting-to-Photos in 3D Reconstructed Visual Reality. In *VRST 2018: 24th ACM Symposium on Virtual Reality Software and Technology (VRST '18), November 28-December 1, 2018, Tokyo, Japan.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3281505.3281528

Figure 1: An example interface of orbiting through a set of photos that observe a point-of-interest in a 3D reconstructed scene. This paper investigates different user interfaces for such orbiting-to-photos techniques. Here, blue cameras indicate possible photos to move to, while the currently viewed photo is shown in the center of the screen. The red orbit figure is for illustration only and does not appear in the interfaces.

1 INTRODUCTION

Photo-based 3D virtual navigation is an exciting technology that enables users to virtually explore real-world scenes. Users can virtually travel through 3D image-based reconstructed models and also visit photos in the scene that were used to create the reconstructed model. Applications include virtually exploring tourist areas [27], viewing street-level photography [1], and remotely collaborating in augmented reality [9, 10]. In this paper, we consider a particular virtual travel maneuver that is important for this type of virtual navigation—that is, orbiting to photos that can see a point-of-interest (POI). Orbiting around a POI in 3D reconstructed visual reality could be used for virtually inspecting historical artifacts, crime scene investigation, virtual field trips, interior decoration, and examining hard-to-visit yet scientifically important areas such as volcanoes, caves, or Mars. We call this paradigm of visiting photos by orbiting about a point-of-interest "orbiting-to-photos."

The main challenge with this particular type of orbiting is how to give appropriate feedback to the user regarding the existence and information of each photo in 3D while allowing the user to manipulate three degrees-of-freedom (DoF) for orbiting around the POI. Previous work in orbiting-to-photos mostly focused on densely captured POIs [26] and small scenes in the context of remote collaboration [9]. In this paper, we utilize desktop-based orbiting interaction since previous photo-based virtual navigation interfaces use this paradigm [9, 26]. We compare three approaches for orbiting-to-photos: two baseline approaches—proxy plane and thumbnails—and our hybrid approach that combines both of them. We present the results from a user study that includes both indoor and outdoor scenes. Our contributions include:

(1) A hybrid orbiting-to-photos virtual travel interface combining proxy plane and thumbnail approaches.
(2) A user experiment comparing two existing approaches with our hybrid method in both densely captured and reconstructed indoor scenes as well as sparsely captured and reconstructed outdoor scenes.

VRST '18, November 28-December 1, 2018, Tokyo, Japan

ACM ISBN 978-1-4503-6086-9/18/11…$15.00
https://doi.org/10.1145/3281505.3281528

2 RELATED WORK

In this section, we focus on prior work related to orbiting; for interested readers, surveys on virtual navigation in general can be consulted [5, 14, 16]. Orbiting-to-photos is related to orbiting movement in both synthetic scenes and image-based reconstructions. Orbiting movement is sometimes categorized under "targeted" or "point-of-interest" movement [16, 22] and may also be called "object-based assistance" [5].

The term "orbiting" is sometimes mixed up with the term "rotating." In this paper, "rotation" refers to when the virtual camera's world position stays constant and its roll, pitch, and/or yaw is modified only; this is akin to turning your head without moving your position. "Orbiting," on the other hand, refers to constrained movement such that the virtual camera's optical axis always intersects a point-of-interest in front of the camera; this is akin to observing a statue from different perspectives[1]. This constraint typically reduces movement to have only two or three degrees-of-freedom: two for determining the angle of viewing perspective towards the POI and one for the distance towards the POI[2].

2.1 Orbiting in Synthetic Scenes

Various orbiting movement techniques have been presented in the past for synthetic virtual environments [4, 7, 13, 17, 22, 24, 25, 28, 30]. HoverCam [17] and more recently SHOCam [24] focus on smoothly orbiting around generic 3D objects, rather than a single 3D point in space. Navidget [13] introduced a widget-based approach to quickly move to a point-of-interest and orient the viewpoint around it. All of these prior works, however, assume a continuous set of equally important viewpoints available to move to, whereas orbiting-to-photos is specifically concerned with orbiting to a discrete set of target viewpoints (*i.e.*, this discrete set, being photos, has higher importance than all other viewpoints).

2.2 Orbiting to Visit Photos in 3D

In terms of orbiting movement for photo-based virtual navigation, only several have explored this in the past [2, 9, 26].

Snavely *et al.* [26] focused on orbiting around popular outdoor landmarks with 2-DoF orbiting. Our investigation in this paper, on the other hand, is concerned with both indoor and outdoor generic points of interest that may or may not have many photos observing them, including POIs in emerging scene reconstructions. Snavely *et al.* also focused on automatically finding good orbits and filtered out photos based on aesthetic qualities; the end result was a constrained movement in an automatically found orbit circle around the POI. No 3-DoF orbiting was reported and no user study was conducted on the usability of their interface. Microsoft Photosynth[3] used a similar orbiting approach but limited orbiting to only 1-DoF (*i.e.*, side to side). The recent technical preview edition of Photosynth also had a version of orbiting that was designed for photos captured in a specific, evenly spaced way to make the final 1-DoF navigation experience more visually pleasing. In this paper, we make no assumptions about how users took photos, and thus our orbiting-to-photos interface is applicable to all types of unstructured captures.

Table 1: Comparison of approaches in terms of degrees-of-freedom (DoF) of movement and user experiments. Previous approaches only had 1-DoF or 2-DoF orbiting, whereas our method allows for 3-DoF movement.

Method	1-DoF	2-DoF	3-DoF	User Exp.
Snavely *et al.* [26]	✓	✓		
Photosynth	✓			
Gauglitz *et al.* [9]	✓			✓
This paper	✓	✓	✓	✓

Gauglitz *et al.* [9] implemented an orbiting-to-photos interface which had 1-DoF multi-touch orbiting with an automatic snapping movement. They used a red-line in-situ visualization along with a thumbnail image to preview the photo to move to. A qualitative user study found that their approach was useful but had some limitations in its visualization (a red-colored ray indicating the snapping position); in our preliminary studies (Sec. 4), we also found that line visualizations were not found to be useful by most people. Gauglitz *et al.* did not compare their method against other orbiting methods. In this paper, we present a comparative evaluation for several interfaces that involve not only 1-DoF or 2-DoF orbiting movements but full 3-DoF orbiting movements. Preliminary evaluations in our paper also confirmed the limitations of Gauglitz *et al.*'s line visualizations for orbiting.

Finally, when there is a high density of images available (*i.e.*, when much of the lightfield has been captured), image-based rendering approaches may be directly used for orbiting [2, 11, 20]. In this paper, we focus especially on emerging image-based reconstructed scenes, which may be sparsely and incompletely reconstructed, and therefore, sophisticated image-based rendering methods are not always directly applicable. Instead, we choose the simple approaches of using proxy planes and thumbnail images, both of which are simple, are applicable to emerging mixed-reality scenes, and have been used in previous related work [8, 26] (see Section 3.3).

3 ORBITING-TO-PHOTOS

We implemented several interfaces for orbiting-to-photos based on a viewpoint selection algorithm, visualizing available photo poses (*i.e.*, cameras), and visualizing a camera's photo in-situ or as a thumbnail. Due to time constraints, we focused on a desktop mouse based implementation; however, a multi-touch implementation would be straightforward. Mouse controls are a straight-forward vanilla orbiting implementation. Left-click dragging modifies the 2-DoF orientation around the POI (pitch and yaw); specifically, left/right movement rotates the camera about a vertical axis parallel to gravity that includes the POI, while up/down movement rotates the camera around a horizontal axis parallel to the camera's right-vector (screen horizontal) that includes the POI. Scrolling in/out moves the camera towards/away from the POI. Double left-clicking updates the orbit center POI by choosing the 3D point underneath or

[1] Note that Jankowski *et al.* [16] equate the terms "rotate" and "orbit," and they instead use "look around" for what we call "rotation" in this paper.
[2] Technically, a fourth degree-of-freedom could be camera roll, but in practice this is typically not used.
[3] https://blogs.msdn.microsoft.com/photosynth/2017/02/06/microsoft-photosynth-has-been-shut-down/.

behind the mouse cursor; for the experiments, we used pre-defined orbit centers.

3.1 Viewpoint Selection Algorithm

During empirical testing of orbiting-to-photos, we realized that predictability of movement is very important for orbiting-to-photos virtual navigation. We considered using several generic viewpoint similarity functions [9, 18, 19, 26, 27], but after some testing, we opted to use a simple function that allows users to easily understand and predict what the system will offer as a photo to snap to. Specifically, we choose a photo $\hat{p}$ from the set of available photos $\mathcal{P}_O$ that can see the 3D orbiting point-of-interest O to snap to via:

$$\hat{p} = \arg\min_{p \in \mathcal{P}_O} \left(\alpha \frac{1}{||O - V||_2} \frac{1}{||O - P||_2} (O - V)^T (O - P) \right) \quad (1)$$

where P is the 3D position of the photo p, V is the position of the virtual camera viewpoint, and α is a temporal consistency adjuster that equals 0.5 when that photo was chosen previously as $\hat{p}$ and 1 otherwise. This essentially returns the photo with the least angular difference between it and the virtual viewpoint with respect to the POI O. Interfaces A & B in the main study used this viewpoint selection algorithm.

3.2 Available Camera Viewpoint Visualizations

3.2.1 Cameras. We experimented with several different visualizations to show the available photos to the user. To distinguish between showing the actual photo and its camera pose, in this section, we simply use the term camera for the photo's position and orientation in 3D space. To help users understand that these photos were taken from a specific location with a specific orientation, we chose to use a simple DSLR camera 3D model as the visualization.

If the camera is in the virtual viewpoint's field of view, we simply show the camera as is. However, if it is outside the field of view, we move it towards the POI until it reaches the edge of the current virtual view frustum. We rescale the out of view cameras to make them a constant size when at the edge of the screen as shown in Figures 2e, 3b, and 3c; however, future work could incorporate off-screen visualization techniques [3, 12] that visualize distances to the off-screen object.

We color the cameras blue to allow users to easily spot the cameras in the scene. When a specific camera's photo is being shown (see Sec. 3.3), we change the color to be red instead (*e.g.*, Figure 3b). Finally, in some situations we fade out the cameras over 1 second to make them invisible to help avoid visual clutter from the interface; Interface B takes this approach as described later in Section 4.2.1.

3.2.2 Lines. We further introduced white lines that were connected from each camera to the POI. An example is shown in Figure 2b. On the one hand, having such lines may be beneficial since it could help users know where each camera is; on the other hand, these lines introduce visual clutter which may not be beneficial. In the end, our preliminary pilot studies found that the lines were confusing to users and some did not see their benefit; thus, our main study did not include any such line visualizations.

3.3 Photo Visualizations

We take the simple approaches of using proxy planes and thumbnail images, both of which are simple, are applicable to emerging mixed-reality scenes, and have been used in previous related work [8, 26].

3.3.1 Proxy Plane. To visualize the photo in the field of view, we use a proxy plane whose normal is parallel to the photo's optical axis and that is positioned to contain the POI [26]. Examples are shown in Figures 2a and 3a. When the chosen photo changes, we quickly fade in the new $\hat{p}$ and fade out the previous photo, using their respective proxy planes. For Interface B that shows the blue camera visualizations, we update the photo and proxy plane whenever the user mouses over a camera visualization; if the proxy plane is greater than 60° away from the current viewpoint, we instead show a thumbnail photo as described in the next section.

3.3.2 Thumbnail. Thumbnail photos can be shown when mousing over the camera visualizations. Examples are shown in Figures 2e and 3c. We make the thumbnail appear to the top-right of the camera visualization, unless it would move outside of the screen and we adjust accordingly.

3.4 Moving to Photos

Finally, users can move to specific photos (*i.e.*, assume its pose) by left-clicking anywhere in the scene. The photo moved to is either the one chosen by the viewpoint selection algorithm or the one represented by the camera visualization that was last moused-over. A subsequent single left-click will undo that movement, bringing the user back to his or her original viewpoint pose before moving to the photo pose.

After moving to the photo, either the POI can still be at the center of the screen or the photo be at the center of the screen; we make this an option in the interface. For the main study, we chose to make the photo the center of the screen to avoid any confusion with clicking to move to photos. In this situation, if the user does not undo the movement to the photo, any left-click dragging will no longer orbit around the POI in the center of the screen. To adjust for this, when the user begins left-click dragging, we first quickly animate the virtual viewpoint to rotate towards the POI again and then resume orbiting movement around the POI. We settled on a 0.5 second animation duration and used this approach during the main user study.

Finally, we also experimented with automatic snapping where if users pause for 0.5 seconds, the view is automatically animated to the photo chosen by the viewpoint selection algorithm. Preliminary testing found that some users did not like automatic snapping, whereas others found it useful but wanted it to be able to toggle automatic snapping on or off. To make things simple for the main study, we decided to leave out this automatic snapping feature entirely.

4 EXPERIMENTS

We first iterated through several different user interfaces for orbiting, inspired by previous work. After running preliminary pilot studies on these interfaces (Section 4.1), we conducted the main study on three interfaces (Section 4.2).

Table 2: Comparison of features in interfaces used in the main study. In essence, interface A (ProxyPlane)'s feature set is the complement of interface C (Thumbnails)'s feature set, while interface B (Hybrid) is a combination of the two.

Feature	A: P. Plane	B: Hybrid	C: Thumb.
Photo on Proxy Plane	✓	✓	
Photo as Thumbnail		✓	✓
Camera Visualizations		✓	✓
Photo updated on rotation	✓	✓	
Photo updated on mouse-over		✓	✓

4.1 Preliminary Pilot Studies

We ran two preliminary pilot studies, each with five participants. We briefly report the major findings here.

The first pilot study had five interfaces as shown in Figure 2. We gave users the tasks of general point-of-interest object inspection and moving to specific photos. Participants experienced two datasets, an indoor densely reconstructed model and an outdoor sparsely reconstructed model. Based on survey results and talking with the participants, we concluded that the fourth red line visualization interface was the weakest ranking one of the five. Furthermore, users did not appreciate as much the full visualization approach that showed many white lines. Similar to Gauglitz *et al.*'s results [9], users did not see much value in the line visualizations.

The second preliminary pilot study focused on three interfaces: the first and last interfaces from the previous pilot study (2a and 2e) were reused, and the third interface was a modified version of the previous pilot study's third interface. Specifically, we eliminated most of the lines and only showed the red line and one white line showing the next "closest" camera after the red highlighted one. Again, however, several users did not see the usefulness of these line visualizations. Thus, for the main study, we simplified the interfaces as described in the next section.

4.2 Main Study

4.2.1 Conditions. For the main study, we used three interfaces label "A," "B," and "C" to not introduce any bias in the participants' responses. A summary of the major differences between the interfaces is shown in Table 2. The three interfaces are shown in Figure 3 and are described next:

(1) Interface A shows photos projected onto a proxy plane & updated based on the viewpoint selection algorithm.
(2) Interface B is essentially a combination of A and C. Photos are shown either on the proxy plane or as a thumbnail if the proxy plane is greater than 60° away from the current viewpoint. Since the photo is shown on the proxy plane in the field of view, we fade out the camera visualizations over 1 second to avoid occluding the in-situ photo. For indoor scenes with many cameras, we make the camera visualizations semi-transparent when they appear inside or near the thumbnail photo.
(3) Interface C has thumbnail photos shown when mousing-over camera visualizations [8]. For indoor scenes with many cameras, we make the camera visualizations semi-transparent when they appear inside or near the thumbnail photo.

Table 3: The number of visitable photos in each dataset, the number of photos observing the POI for each task in the model, and the minimum and maximum distances (in meters) to the POI of all the photos observing the POI.

		Training task			Actual task		
Dataset	$\lvert\mathcal{P}\rvert$	$\lvert\mathcal{P}_O\rvert$	Min	Max	$\lvert\mathcal{P}_O\rvert$	Min	Max
Indoor1	571	47	0.932	3.633	61	0.995	2.780
Indoor2	566	71	1.068	4.111	43	1.003	1.943
Indoor3	256	19	1.302	2.829	29	1.357	3.948
Outdoor1	140	22	1.153	33.190	35	5.806	31.793
Outdoor2	129	21	3.958	33.931	18	3.550	32.007
Outdoor3	219	46	2.786	43.394	23	2.586	38.026

Users experienced the interfaces in two scene types—indoor and outdoor. To avoid bias in acquiring spatial knowledge of the scene, we used three separate indoor and three separate outdoor scenes, for a total of six scenes. Table 3 shows statistics of these models.

4.2.2 Task & Metrics. The major goal of orbiting is typically object inspection. Thus, we gave participants the task of inspecting a pre-defined POI by visiting photos in each scene. POIs were chosen such that a large amount of photos observed it from several different perspectives. Movement was constrained to stay orbiting around the POI and participants inspected each POI for 90 seconds.

To measure how well users could accomplish this task, we collected two quantitative metrics as well as qualitative survey responses. The quantitative metrics included number of photos visited per scene—an indication of how much information the participant was exposed to—and whether or not participants could recall the POI accurately—a more direct measurement of scene understanding. This latter recall task was achieved by showing participants two photos of the POI at a later time: one real and one fake altered photo. The real photo was altered slightly and both photos blurred to increase task difficulty. After seeing both photos, we asked participants to tell us which one they believed was the real photo; see Figure 4 for an example pair of photos.

4.2.3 Procedure. Participants first completed a pre-study questionnaire to gather demographic information. After a brief introduction to the study, users experienced three indoor scenes and then three outdoor scenes (or vice versa to balance the order). With each scene, participants used one interface (order counterbalanced) and were trained on the interface before performing the actual task. After the task, users completed a short questionnaire and then we had participants complete the recall task. After each scene type (indoor or outdoor), users also completed a brief questionnaire. Finally, after all the conditions, users completed a post-study questionnaire.

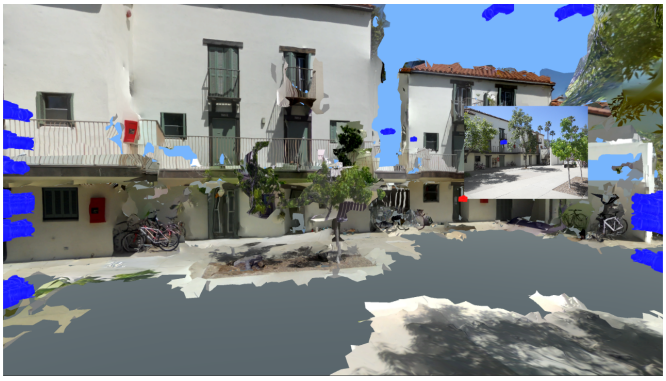
4.3 Results

4.3.1 Participants. 12 participants completed the study (avg. 19.75 years old, min 19, max 21). 5 were male and 7 female; there were no statistically significant results based on gender in the results reported in the following sections. 5 almost never used a computer mouse regularly, 4 several days a week, and 3 every day. 5 stated being not familiar with interactive 3D software, 6 barely familiar, and 1 somewhat familiar. On average, they rated their skills with

(a) Only proxy plane. (b) All visualizations. (c) Fading visualizations. (d) Red line visualization. (e) Thumbnails.

Figure 2: Five different interfaces tested in the first preliminary pilot study: (a) photo shown on a proxy plane in the field of view [26]; (b) photo shown on proxy plane with white lines going from the POI to each camera visualizations; (c) a fading version of (b); (d) thumbnail preview in the corner with a red line visualization [9]; (e) thumbnails shown when mousing over a camera visualization [8].

(a) Interface A: Only proxy plane. (b) Interface B: Hybrid.



(c) Interface C: Thumbnails.

Figure 3: Three interfaces used for the main study: (a) photo shown on a proxy plane in the field of view [26]; (b) hybrid / combination of (a) and (c); (c) thumbnails shown when mousing over a camera visualization [8].

3D software as 1.667 on a 7-point Likert scale, where 1 is Novice and 7 is Expert.

We also used the Santa Barbara Sense of Direction Scale (SBSOD) [15] to assess participants' spatial abilities. Participants had a median overall SBSOD score of 4.967 (avg. 4.644, min 2.8, max 5.667). For further analysis, we divided participants into two equally-sized groups (6 in each), those with a SBSOD of less than 5 and those with 5 or above.

4.3.2 Inspection Task Results. We used a two-way repeated measures ANOVA to analyze the results of how many photos the users saw for each interface and scene. To make the evaluation fair across different datasets with differing numbers of available photos (see Table 3), we used percentages of available photos seen (either by visiting the photo or mousing over for interfaces B and C). There was no main effect for factor interface ($F_{2,22}$ = 0.507, p = 0.609; average percentages 57.61%, 61.03%, 64.61%, respectively for A, B, and C). There was a main effect for factor scene ($F_{1,11}$ = 7.635, p =

(a) Image 1. (b) Image 2.

Figure 4: Which photo is the real one? The answer can be deduced by looking at Figure 3a.

0.019; average percentages 57.59% and 64.57% for indoor and outdoor scenes, respectively). It may be that the smaller amount of photos in the outdoor scenes allowed for a higher percentage of photos to be seen.

4.3.3 After-Interface Questionnaire. Likert statements and results are shown in Figure 5. We used two-way repeated-measures ART ANOVAs [29] to analyze the responses, with factors interface and scene. Participants disagreed with the statement regarding frustration more with interface C than with interface A. Users were more confident with B than A and also felt that they could go to the photos they wanted to see more so with B than A. They also felt that B helped them get a spatial understanding of the point of interest and its immediate surroundings more so than for interface A. Users felt they knew where photos were and where they were more so with B than both A and C.

It is interesting that a large amount of participants disagreed that the interfaces were visually pleasing. For at least interfaces A and B, differences in field of view and scene visibility overlap between different available photos can cause visual distraction. For example, a photo seeing the point-of-interest on the right-hand side of its image will have little scene visibility overlap with a photo that sees the point-of-interest on the left-hand side of its image. Thus, showing in-situ photo visualizations for the former photo and then the latter photo will result in possible visual distractions. Common ways to avoid such distracting in-situ photo visualizations include: drawing a bounding box around the point-of-interest so that the photo is seen only inside the box [26] and of course, not showing such in-situ photo visualizations (as in interface C). Still, it is interesting that interface B had the highest agreement with this statement, since interface B was the most complex visually.

It is also interesting that the proxy plane approach that always showed the photo in-situ had the least agreement on the statement regarding getting a spatial understanding of the POI and its immediate surroundings. The hybrid approach (interface B) was rated statistically higher than the proxy plane approach for this statement. It could be that having the visualizations of the cameras help users understand the spatial context of the POI with its surroundings.

We also found that users reporting a higher SBSOD score had a higher reported confidence, using a two-way mixed factorial ART ANOVA ($F_{1,10}$ = 5.849, p = 0.036). Figure 6 shows the results. It is logical that those with a higher reported sense of direction would feel more confident in virtually navigating these reconstructed scenes.

There were also several interaction effects. The statement "I felt frustrated using this interface" had an interaction effect between factors scene and interface ($F_{2,22}$ = 3.942, p = 0.034). For indoor scenes, there was a main effect by interface ($F_{2,22}$ = 9.466, p = 0.001), with pairwise comparisons showing A > C (p < 0.001). Figure 7a shows the results. This may be due to the fact that outdoor scenes spanned large spaces and thus some photos that could see the POI were very far away from the POI. In turn, this may have made it harder to select photos by mousing over smaller camera visualization that were far away. In addition, if the user selected a far away photo, it would be harder to get back closer to the POI if there were not any nearby camera visualizations.

There was also an interaction effect for the results for the statement "This interface helped me get a spatial understanding of the point of interest and its immediate surroundings" ($F_{2,22}$ = 11.569, p < 0.001). For outdoors, there was a main effect by factor interface ($F_{2,22}$ = 12.447, p < 0.001), with pairwise comparisons showing that B > A (p < 0.001) and B > C (p = 0.005). Figure 7b shows the results. It may be that the hybrid approach gives the highest ranking for this statement outdoors since users also felt it allowed them to know what photos were available and where they were (see Likert response results in Figure 5).

Finally, there was also an interaction effect for the results for the statement "This interface made me feel like I was actually in the scene" ($F_{2,22}$ = 3.933, p = 0.035). For outdoors, there was a main effect by factor interface ($F_{2,22}$ = 6.693, p = 0.005), with pairwise comparisons showing that B > A (p = 0.026) and B > C (p = 0.007). Figure 7c shows the results. We are not fully sure why the hybrid method would cause the highest agreement with this statement for outdoor scenes. We originally hypothesized that the proxy plane method would have the highest agreement since it does not show any camera visualizations. Thus, it is surprising that the addition of the camera visualizations causes users to feel more like being in the scene. Future work should investigate this further; it could be that knowing where photos were taken gives users more of a sense of the space and thus makes them feel more like being in the scene.

4.3.4 Recall Task. We analyzed the results for the recall task using repeated-measures logistic regression. Figure 8 shows the results. While there appears to be a trend towards C having the highest accuracy, the results were non-significant. Future work should investigate other spatial understanding or recall tasks.

4.3.5 After-Scene Questionnaire. Overall rankings are shown in Figure 9a. Using a two-way repeated-measures ART ANOVA, the overall preference rankings were statistically significant ($F_{2,22}$ = 3.598, p = 0.044) with pairwise comparisons showing B ranked higher than A (p = 0.036). Unfortunately, the F-values on the aligned responses not of interest were close but not equal to 0, so the ART ANOVA method may not be reliable. Therefore, to be safe, we also ran a one-way repeated-measures ART ANOVA on only the factor interface. The F-values on the aligned responses not of interest were now equal to 0, and the result was statistically significant ($F_{2,22}$ = 3.804, p = 0.038) with pairwise comparisons showing B ranked higher than A (p = 0.030).

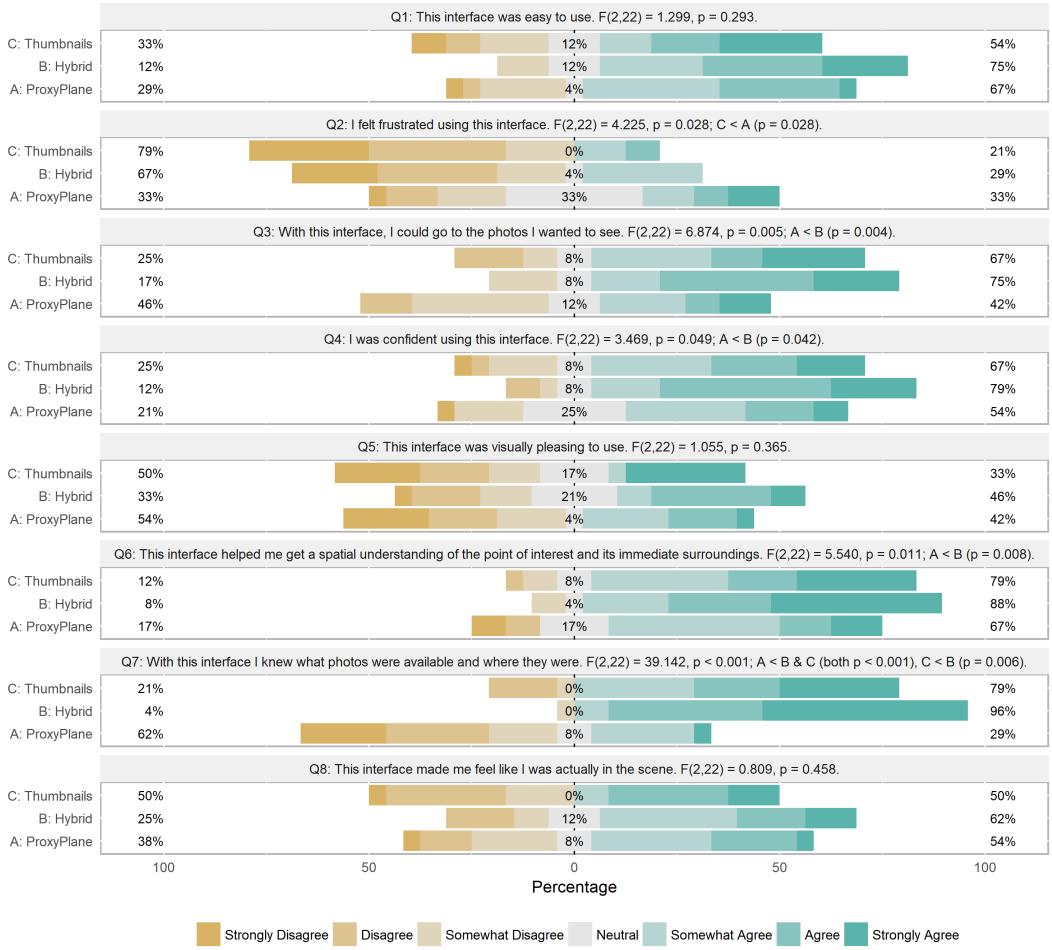


Figure 5: After-Interface Questionnaire Results (best viewed in color). Percentages on the left show total for all disagreement responses, percentages in the middle show neutral responses, and percentages on the right show total for all agreement responses.

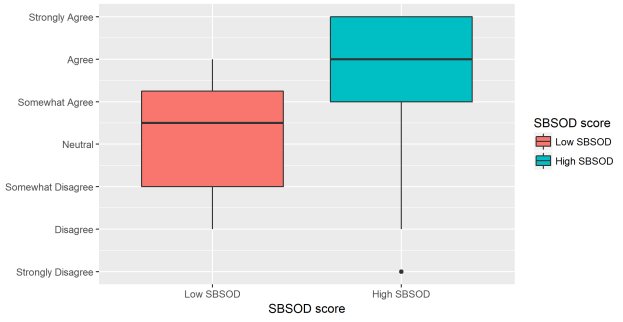
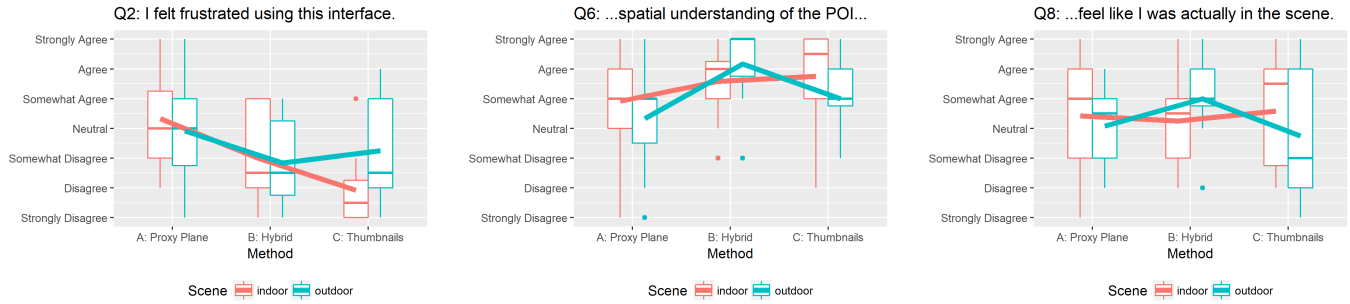


Figure 6: Confidence responses based on SBSOD scores greater than or equal to 5 or less than 5.

There was also an interaction effect ($F_{2,22}$ = 4.430, p = 0.024) between interface and scene for preference rankings. For outdoor scenes, there was a main effect by factor interface ($F_{2,22}$ = 11.31, p < 0.001), with pairwise comparisons indicating B > A (p < 0.001) and B > C (p = 0.028). Figure 9 shows the results. It is interesting that for outdoor scenes, our hybrid approach (interface B) was ranked higher than both baselines. It may be that having both controls of orbiting to update the photo and mousing-over camera visualizations to update the photo was more necessary for outdoor scenes than for indoor scenes. Therefore, we can say that users preferred our hybrid method over both baselines for outdoor scenes.

We also analyzed the responses to 4 Likert statements using a one-way repeated measures ART ANOVA. The results are shown in Figure 10. Only the statement regarding the camera visualizations being useful was statistically significant, with users agreeing with that statement more for outdoor scenes than indoor scenes.

Using a two-way mixed factorial ART ANOVA we also found an interaction effect between scene and SBSOD score ($F_{1,10}$ = 14.134, p = 0.004) for the statement, "The (blue) camera visualizations in Interfaces B & C are useful." For lower SBSOD scores, there was a



(a) Frustration. **(b) Spatial Understanding.** **(c) Feel like being there.**

Figure 7: After-Interface Questionnaire interaction effects between method and scene. The solid lines connect the mean values for each condition.

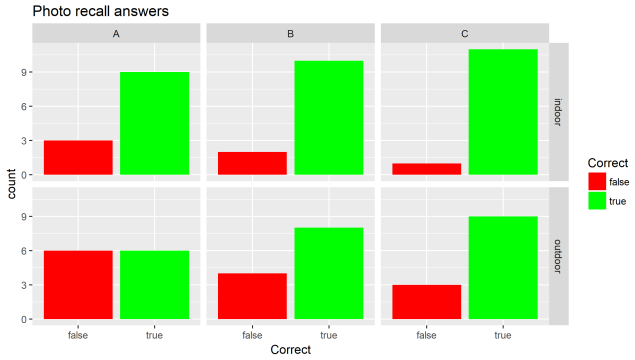


Figure 8: Photo recall task results.

main effect by factor scene ($F_{1,5}$ = 8.767, p = 0.031), where indoor responses agreed less with this statement than outdoor responses. Figure 11 shows the results.

4.3.6 Post-Study Questionnaire. We used a one-way repeated measures ART ANOVA to analyze the rankings and there was no statistical significance ($F_{2,22}$ = 1.092, p = 0.353). The rankings are shown in Figure 12.

5 DISCUSSION

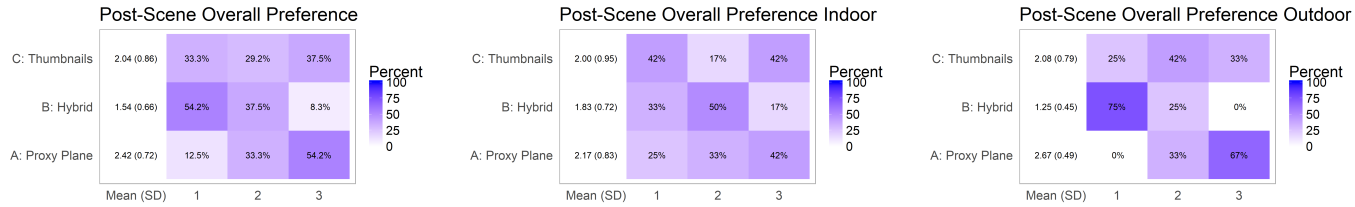
In this paper, we evaluated three orbiting-to-photos interfaces. We note that the general concept of orbiting-to-photos may also be applied to orbiting to bookmarked views in synthetic scenes [6, 8], and therefore our results may be applicable to the more general case of orbiting in synthetic scenes as well. We also focused on unstructured captures of visual reality as is common with mixed-reality remote collaboration scenarios.

In general, we found that keeping interfaces as simple as possible is the best approach for supporting virtual navigation. However, for 3-DoF orbiting-to-photos, it can be hard to keep things simple. For example, letting users know about the available photos introduces a level of complexity to the interface which works against the simplicity principle. On the other hand, not explicitly letting users know about the available photos (*e.g.*, as with Interface A) causes increased frustration. Perhaps the optimal approach is to find a default middle ground (*e.g.*, Interface B's hybrid approach) and then allow users to toggle on and off different features according to their choosing (*e.g.*, showing camera visualizations).

It is interesting that with the constrained virtual navigation scenario of orbiting (with only 3-DoF or less), showing camera visualizations becomes a viable option again for a virtual travel interface. This was possible by filtering out which cameras could actually see the point of interest in the scene, thus alleviating the visual clutter problem that essentially renders this type of interface less useful or not usable at all. It was also interesting that users agreed more spatial understanding statement for our hybrid interface than for the proxy plane approach. This may indicate that having the additional blue camera visualizations not only helps in terms of virtual travel but also in terms of wayfinding (*i.e.*, the cognitive aspect of virtual navigation). Future work should investigate this possibility further.

This paper also highlighted the importance of viewpoint visualizations. When constraining the navigation and allowing users to select photos to visit, visualizations become a very important feedback mechanism which helps users in interacting with the scene. While our hybrid approach was ranked the highest for outdoor scenes and better than the proxy plane baseline overall, there is still much room for improvement, especially in the area of visual aesthetics (as noted in the Likert statement responses). Future work should include using more sophisticated image-based rendering instead of proxy plane geometry and thumbnail images. Early work in light-field rendering included examples of 2-DoF orbiting around an object. Extending this to the more general case of 3-DoF orbiting and especially for emerging scene reconstructions may be challenging.

Future work should also conduct user experiments where participants can choose their own orbit centers. We could then see how they use orbiting-to-photos in conjunction with more general exploration. For emerging scene reconstructions, would orbiting be used more than other travel techniques, such as walking or flying metaphors?



(a) Overall. **(b) Indoors.** **(c) Outdoors.**

Figure 9: After-Scene preference rankings overall (a) and subdivided by scene type (b-c).

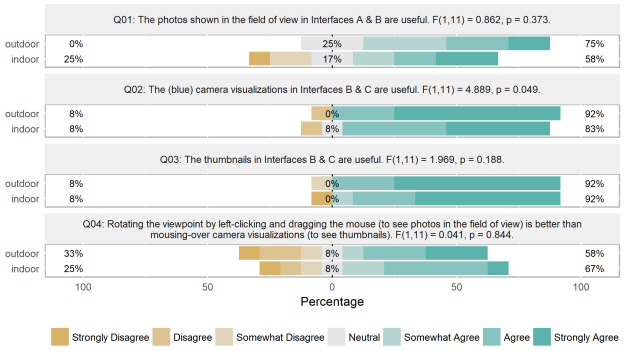


Figure 10: After-Scene Likert statement responses.

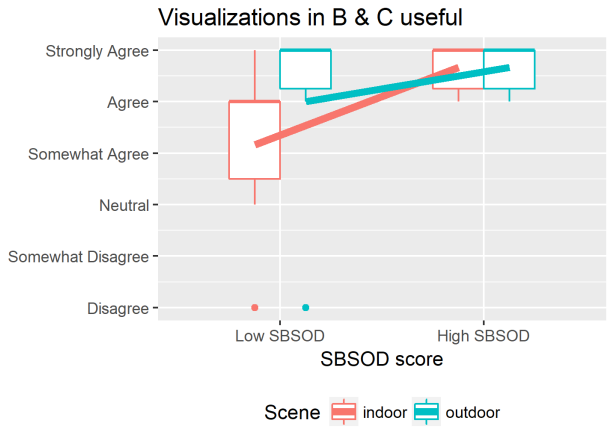


Figure 11: Camera visualization responses based on SBSOD scores and scene for the statement "The (blue) camera visualizations in Interfaces B & C are useful." Solid lines connect the means for each condition.

Another avenue to explore is if these results change for a multi-touch version. One major change would involve redesigning mouse-over photo updates since multi-touch UIs typically do not have a mouse-over interaction paradigm.

Finally, another interesting idea for future work is to utilize semantic segmentation [21, 23] of the image-based reconstruction to orbit around automatically detected and segmented objects, rather than simple 3D points. This would allow one to apply methods such as HoverCam [17] and SHOCam [24] to guide movement around the object. In this case, choosing a more appropriate viewpoint selection algorithm for visiting photos may be an interesting challenge.

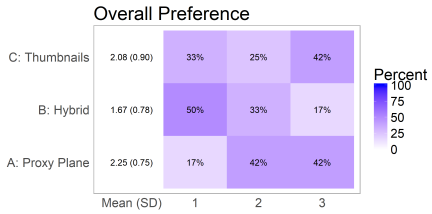


Figure 12: Post-Study interface rankings.

ACKNOWLEDGMENTS

This work was supported by NSF grant IIS-1423676 and ONR grant N00014-16-1-3002.

REFERENCES

[1] Dragomir Anguelov, Carole Dulong, Daniel Filip, Christian Frueh, Stphane Lafon, Richard Lyon, Abhijit Ogale, Luc Vincent, and Josh Weaver. 2010. Google street view: Capturing the world at street level. *Computer* 43, 6 (2010), 32–38. https://doi.org/10.1109/MC.2010.170

[2] A. Arpa, L. Ballan, R. Sukthankar, G. Taubin, M. Pollefeys, and R. Raskar. 2013. CrowdCam: Instantaneous Navigation of Crowd Images Using Angled Graph. In *2013 International Conference on 3D Vision - 3DV 2013*. 422–429. https://doi.org/10.1109/3DV.2013.62

[3] Patrick Baudisch and Ruth Rosenholtz. 2003. Halo: A Technique for Visualizing Off-screen Objects. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '03)*. ACM, New York, NY, USA, 481–488. https://doi.org/10.1145/642611.642695

[4] Michael Chen, S. Joy Mountford, and Abigail Sellen. 1988. A Study in Interactive 3-D Rotation Using 2-D Control Devices. *SIGGRAPH Comput. Graph.* 22, 4 (June 1988), 121–129. https://doi.org/10.1145/378456.378497

[5] Marc Christie, Patrick Olivier, and Jean Marie Normand. 2008. Camera control in computer graphics. *Computer Graphics Forum* 27, 8 (dec 2008), 2197–2218. https://doi.org/10.1111/j.1467-8659.2008.01181.x

[6] M. Di Benedetto, F. Ganovelli, M. Balsa Rodriguez, A. Jaspe Villanueva, R. Scopigno, and E. Gobbetti. 2014. ExploreMaps: Efficient construction and ubiquitous exploration of panoramic view graphs of complex 3D environments. *Computer Graphics Forum* 33, 2 (may 2014), 459–468. https://doi.org/10.1111/cgf.12334

[7] George Fitzmaurice, Justin Matejka, Igor Mordatch, Azam Khan, and Gordon Kurtenbach. 2008. Safe 3D Navigation. In *Proceedings of the 2008 Symposium on Interactive 3D Graphics and Games (I3D '08)*. ACM, New York, NY, USA, 7–15. https://doi.org/10.1145/1342250.1342252

[8] Thomas Forgione, Axel Carlier, Géraldine Morin, Wei Tsang Ooi, and Vincent Charvillat. 2016. Impact of 3D Bookmarks on Navigation and Streaming in a Networked Virtual Environment. In *Proceedings of the 7th International Conference on Multimedia Systems (MMSys '16)*. ACM, New York, NY, USA, Article 9, 10 pages. https://doi.org/10.1145/2910017.2910607

[9] Steffen Gauglitz, Benjamin Nuernberger, Matthew Turk, and Tobias Höllerer. 2014. In Touch with the Remote World: Remote Collaboration with Augmented Reality Drawings and Virtual Navigation. In *Proceedings of the 20th ACM Symposium on Virtual Reality Software and Technology (VRST '14)*. ACM, New York, NY, USA, 197–205. https://doi.org/10.1145/2671015.2671016

[10] Steffen Gauglitz, Benjamin Nuernberger, Matthew Turk, and Tobias Höllerer. 2014. World-stabilized Annotations and Virtual Scene Navigation for Remote Collaboration. In *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology (UIST '14)*. ACM, New York, NY, USA, 449–459. https://doi.org/10.1145/2642918.2647372

[11] Steven J. Gortler, Radek Grzeszczuk, Richard Szeliski, and Michael F. Cohen. 1996. The lumigraph. *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques - SIGGRAPH '96* (1996), 43–54. https://doi.org/10.1145/237170.237200

[12] Sean Gustafson, Patrick Baudisch, Carl Gutwin, and Pourang Irani. 2008. Wedge: Clutter-free Visualization of Off-screen Locations. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '08)*. ACM, New York, NY, USA, 787–796. https://doi.org/10.1145/1357054.1357179

[13] M. Hachet, F. Decle, S. Knodel, and P. Guitton. 2008. Navidget for Easy 3D Camera Positioning from 2D Inputs. In *2008 IEEE Symposium on 3D User Interfaces*. 83–89. https://doi.org/10.1109/3DUI.2008.4476596

[14] Chris Hand. 1997. A Survey of 3D Interaction Techniques. *Computer Graphics Forum* 16, 5 (1997), 269–281. https://doi.org/10.1111/1467-8659.00194 arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1111/1467-8659.00194

[15] Mary Hegarty, Anthony E Richardson, Daniel R Montello, Kristin Lovelace, and Ilavanil Subbiah. 2002. Development of a self-report measure of environmental spatial ability. *Intelligence* 30, 5 (2002), 425 – 447. https://doi.org/10.1016/S0160-2896(02)00116-2

[16] J. Jankowski and M. Hachet. 2015. Advances in interaction with 3D environments. *Computer Graphics Forum* 34, 1 (2015), 152–190. https://doi.org/10.1111/cgf.12466

[17] Azam Khan, Ben Komalo, Jos Stam, George Fitzmaurice, and Gordon Kurtenbach. 2005. HoverCam: Interactive 3D Navigation for Proximal Object Inspection. In *Proceedings of the 2005 Symposium on Interactive 3D Graphics and Games (I3D '05)*. ACM, New York, NY, USA, 73–80. https://doi.org/10.1145/1053427.1053439

[18] Johannes Kopf, Michael F. Cohen, and Richard Szeliski. 2014. First-person Hyperlapse Videos. *ACM Trans. Graph.* 33, 4, Article 78 (July 2014), 10 pages. https://doi.org/10.1145/2601097.2601195

[19] Avanish Kushal, Ben Self, Yasutaka Furukawa, David Gallup, Carlos Hernandez, Brian Curless, and Steven M. Seitz. 2012. Photo tours. *Proceedings - 2nd Joint 3DIM/3DPVT Conference: 3D Imaging, Modeling, Processing, Visualization and Transmission* (Oct. 2012), 57–64. https://doi.org/10.1109/3DIMPVT.2012.62

[20] Marc Levoy and Pat Hanrahan. 1996. Light Field Rendering. In *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '96)*. ACM, New York, NY, USA, 31–42. https://doi.org/10.1145/237170.237199

[21] Kuo-Chin Lien, Benjamin Nuernberger, Tobias Höllerer, and Matthew Turk. 2016. PPV: Pixel-Point-Volume Segmentation for Object Referencing in Collaborative Augmented Reality. In *2016 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. 77–83. https://doi.org/10.1109/ISMAR.2016.21

[22] Jock D. Mackinlay, Stuart K. Card, and George G. Robertson. 1990. Rapid Controlled Movement Through a Virtual 3D Workspace. *SIGGRAPH Comput. Graph.* 24, 4 (Sept. 1990), 171–176. https://doi.org/10.1145/97880.97898

[23] John McCormac, Ankur Handa, Andrew Davison, and Stefan Leutenegger. 2017. Semanticfusion: Dense 3d semantic mapping with convolutional neural networks. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*.

[24] Michael Ortega, Wolfgang Stuerzlinger, and Doug Scheurich. 2015. SHOCam: A 3D Orbiting Algorithm. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology (UIST '15)*. ACM, New York, NY, USA, 119–128. https://doi.org/10.1145/2807442.2807496

[25] Ken Shoemake. 1992. ARCBALL: A User Interface for Specifying Three-dimensional Orientation Using a Mouse. In *Proceedings of the Conference on Graphics Interface '92*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 151–156. http://dl.acm.org/citation.cfm?id=155294.155312

[26] Noah Snavely, Rahul Garg, Steven M. Seitz, and Richard Szeliski. 2008. Finding Paths Through the World's Photos. *ACM Trans. Graph.* 27, 3, Article 15 (Aug. 2008), 11 pages. https://doi.org/10.1145/1360612.1360614

[27] Noah Snavely, Steven M. Seitz, and Richard Szeliski. 2006. Photo tourism: Exploring Photo Collections in 3D. *ACM Transactions on Graphics* 25, 3 (July 2006), 835. https://doi.org/10.1145/1141911.1141964

[28] Desney S. Tan, George G. Robertson, and Mary Czerwinski. 2001. Exploring 3D Navigation: Combining Speed-coupled Flying with Orbiting. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '01)*. ACM, New York, NY, USA, 418–425. https://doi.org/10.1145/365024.365307

[29] Jacob O. Wobbrock, Leah Findlater, Darren Gergle, and James J. Higgins. 2011. The Aligned Rank Transform for Nonparametric Factorial Analyses Using Only Anova Procedures. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '11)*. ACM, New York, NY, USA, 143–146. https://doi.org/10.1145/1978942.1978963

[30] Robert Zeleznik and Andrew Forsberg. 1999. UniCam—2D Gestural Camera Controls for 3D Environments. In *Proceedings of the 1999 Symposium on Interactive 3D Graphics (I3D '99)*. ACM, New York, NY, USA, 169–173. https://doi.org/10.1145/300523.300546